Universidade do Minho
Escola de Engenharia

Cristina Ferreira Teixeira

Designing screws for polymer compounding in twin-screw extruders

Abril de 2013

Designing screws for polymer compounding in twin-screw extruders
Cristina Ferreira Teixeira
UMinho | 2013

Universidade do Minho
Escola de Engenharia

Cristina Ferreira Teixeira

Designing screws for polymer compounding in twin-screw extruders

Tese de Doutoramento
Ciência e Engenharia de Polímeros e Compósitos

Trabalho efectuado sob a orientação do
Professor Doutor António Gaspar Cunha
Professor Doutor José António Covas

Abril de 2013

What lies behind us and what lies before us
are tiny matters
compared to what lies within us.

- Ralph Waldo Emerson -

We don't meet people by accident.
They are meant to cross our path for a reason.
- Unknown -

Acknowledgments

This work will not be possible without the scientific, academic, financial and personal support of several persons and institutions to whom I would like to take this opportunity to thank.
Firstly, I would like to express my gratitude to my supervisors António Gaspar-Cunha and José António Covas for the opportunity to undertake this project and for their supervision, knowledge, scientific guidance and important and relevant suggestions.

I hereby acknowledge the Portuguese Foundation for Science and Technology (FCT) for their financial support under PhD scholarship SFRH/BD/21921/2005.

I am grateful to all colleagues, researchers and academic, technical and administrative staff of Polymer Engineering Department of the University of Minho which in one way or another aided me along these years. A special thanks to João Paulo Peixoto for his technical assistance during all experimental work in co-rotating twin screw extruder. A particular thanks to Fátima Almeida, Marta Silva, Natália Domingues, Tânia Ferreira, Sílvia Ramôa, Silvia Santos and Filomena Costa for their collaboration, friendship, fruitful discussions and long coffee breaks.

I am also profoundly grateful to Thomas Stützle for the opportunity to work and stay in Institut the Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA) in Université Libre de Bruxelles. Thank you for your knowledge and scientific support, encouragement and availability. To Prasanna Balaprakash, Mohamed Saifullah, Eric, Marco Montes de Oca, Arne Brutschy, Nithin Mathews, Eliseo Ferrante, Carlo Pinciroli, Giovanni Pini and all the members of IRIDIA, who kindly received and accepted me as member of IRIDIA family. Thank you for your help, hospitality and friendship which made my stay and work in Brussels a great and pleasant time.

To Luis Paquete (Department of Informatics Engineering - University of Coimbra) and Manuel López-Ibáñez (IRIDIA) for enlightened explanations about the Empirical Attainment Functions and to provide the tools used to compare the performance of the multi-objective algorithms.

To all my friends who have always been on my side along these years, in good and less good moments. Thank you for all the times you reminded me that I had to finish this thesis. Especially to you: for your love, comprehension, dedication and encouragement. No words can express my gratitude. Thank you for always being around!

Last but not least, to my dear parents for all their love, advices and support along all my life, mainly in this long journey. Thanks for always supporting my decisions!

Thank you all for been part of my life!

Cristina Teixeira

Abstract

Considering its modular construction, co-rotating twin screw extruders can be easily adapted to work with polymeric systems with more stringent specifications. However, their geometrical flexibility makes the performance of these machines strongly dependent on the screw configuration. Therefore, the definition of the adequate screw geometry to use in a specific polymer system is an important process requirement which is currently achieved empirically or using a trial-and-error basis.

The aim of this work is to develop an automatic optimization methodology able to define the best screw geometry/configuration to use in a specific compounding/reactive extrusion operation, reducing both cost and time. This constitutes an optimization problem where a set of different screw elements are to be sequentially positioned along the screw in order to maximize the extruder performance.

For that, a global modeling program considering the most important physical, thermal and rheological phenomena developing along the axis of an intermeshing co-rotating twin screw extruder was initially developed. The accuracy and sensitivity of the software to changes in the input parameters was tested for different operating conditions and screw configurations using a laboratorial Leistritz LSM 30.34 extruder. Then, this modeling software was integrated into an optimization methodology in order to be possible solving the Twin Screw Configuration Problem. Multi-objective versions of local search algorithms (Two Phase Local Search and Pareto Local Search) and Ant Colony Optimization algorithms were implemented and adapted to deal with the combinatorial, discrete and multi-objective nature of the problem. Their performance was studied making use of the hypervolume indicator and Empirical Attainment Function, and compared with the Reduced Pareto Search Genetic Algorithm (RPSGA) previously developed and applied to this problem. In order to improve the quality of the results and/or to decrease the computational cost required by the optimization methodology, different hybrid algorithms were tested. The approaches developed considers the use of local search procedures (TPLS and PLS algorithms) into population based metaheuristics, as MOACO and MOEA algorithms.

Finally, the optimization methodology developed was applied to the optimization of a starch cationization reaction. Several starch cationization case studies, involving different screw elements,

screw lengths and conflicting objectives, were tested in order to validate this technique and to prove the potential of this automatic optimization methodology.

Resumo

Devido à sua construção modular, as extrusoras de duplo-fuso co-rotativas podem ser facilmente adaptadas a sistemas poliméricos que requerem especificações mais rigorosas. No entanto, esta flexibilidade geométrica torna o seu desempenho fortemente dependente da configuração do parafuso.

Por isso, a tarefa de definir a melhor configuração do parafuso para usar num determinado sistema polimérico é um requisito importante do processo que é actualmente realizada empiricamente ou utilizando um processo de tentativa erro.

O objectivo principal deste trabalho é desenvolver uma metodologia automática de optimização que seja capaz de definir a melhor configuração/geometria do parafuso a usar num determinado sistema de extrusão, reduzindo custos e tempo. Este problema é um problema de optimização, onde os vários elementos do parafuso têm que ser sequencialmente posicionados ao longo do eixo do parafuso de forma a maximizar o desempenho da extrusora.

Para isso, foi inicialmente desenvolvido um programa de modelação que considera os mais importantes fenómenos físicos, térmicos e reológicos que ocorrem ao longo da extrusora de duplo fuso co-rotativa. De forma a testar a precisão e a sensibilidade do software às alterações dos parâmetros, diversas condições operativas e configurações de parafuso foram testadas tendo como base uma extrusora laboratorial Leistritz LSM 30.34. Seguidamente, este software de modelação foi integrado numa metodologia de optimização com vista à resolução do problema de configuração da extrusora de duplo-fuso. Para lidar com a natureza combinatorial, discreta e multi-objectiva do problema em estudo, foram adaptadas e implementadas versões multi-objectivas de algoritmos de procura local (Two-Phase Local Search and Pareto Local Search) e Ant Colony Optimization. O desempenho dos diversos algoritmos foi estudado usando o hipervolume e as Empirical Attainment Functions. Os resultados foram comparados com os resultados obtidos com o algoritmo genético Reduced Pareto Search Genetic Algorithm (RPSGA) desenvolvido e aplicado anteriormente a este problema.

Com o objectivo de melhorar a qualidade dos resultados e/ou diminuir o esforço computacional exigido pela metodologia de optimização, foram testadas diversas hibridizações. Os algoritmos

híbridos desenvolvidos consideram a integração de algoritmos de procura local (TPLS e PLS) noutras metheuristicas, como MOACO e MOEA.

Por fim, a metodologia de optimização desenvolvida neste trabalho foi testada na optimização de uma reacção de cationização do amido. Para validar esta técnica e provar o seu potencial, foram realizados vários estudos envolvendo diferentes elementos e comprimentos de parafusos, bem como, a optimização de objectivos em conflito.

Contents

List of Figures

List of Tables

List of Symbols

Greek Characters

α	Staggering angle
$\dot{\gamma}$	Shear rate
δ_C	Thickness of the melt film C
δ_{DE}	Thickness of the melt film DE
Δ_{ij}	Increment on trail (i, j)
Δz	Differential element in the $z-$direction
ε_p	Numerical difference of pressure
η	Viscosity
η_0	Zero-shear viscosity
λ	Weight
λ_i	$i-$Component weight
$\hat{\lambda}$	Heat of fusion
π	Permutation
π_i	$i-$Component of permutation
ρ	Pheromone evaporation ratio
ρ_s	Solid density
ρ_m	Melt density
τ_{ij}	Pheromone value in trail (i, j)
τ_{ini}	Initial pheromone value
$\tau_{\max}$	Maximum limit of pheromone value
$\tau_{\min}$	Minimum limit of pheromone value

Roman Characters

a	Empirical constant of Carreau-Yasuda law
A	Starch
$[A]_0$	Initial concentration of starch

$[A]_t$	Concentration of starch hydroxyl groups at time t
B	Reagent
$[B]_0$	Initial concentration of reagent
$[B]_t$	Concentration of reagent at time t
C	Cationic starch
C	Torque
C_s	Specific heat capacity of the solid polymer
C_m	Specific heat capacity of the melt
D_{ext}	External screw diameter
D_{int}	Internal screw diameter
DS	Degree of substitution
DS_{th}	Maximum degree of substitution
e_r	Restrictive element
E/R	Temperature coefficient of viscosity
f	Objective function
$FP*$	Pareto Front
H	Channel height
$H_{\max}$	Maximum channel height
H_m	Melting heat
k	Maximum number of ants used to update the pheromone matrix
k_s	Thermal conductivity of the solid polymer
k_m	Thermal conductivity of the melt
loc	Number of screw elements whose location must be defined
L_o	Initial screw location
m	Number of ants
n	Empirical constant of Carreau-Yasuda law
N	Screw speed
N	Number of individuals of internal population (RPSGA)
N_{ranks}	Pre-defined number of ranks
$N(s)$	Neighborhood of solution s

p	Number of colonies
p_i	Internal population
p_e	External population
${p_{ij}}^m$	Probability of ant m to choose trail (i, j)
P	Pressure
$P*$	Pareto Optimal Set
q_b	Heat flux on the barrel
q_s	Heat flux on the root of the screws
q_f	Heat flux on the flights
Q	Volumetric output
Q_C	Flow through a C-shaped chamber
Q_L	Leakage flow in kneading discs
Q_{starch}	Starch feed rate
RE	Reaction efficiency
r	Rank number
t	Time
T	Temperature
T_0	Reference temperature
T_m	Melting temperature
T_{so}	Temperature of solid polymer at extruder entrance
T_s	Screw surface temperature
T_b	Barrel temperature
T_{Max}	Maximum temperature
$T(y)$	Cross temperature profile (direction y)
V_b	Barrel velocity
V_{bx}	Component x of barrel velocity
V_{bz}	Component z of barrel velocity
V_{sz}	Solid bed velocity
W	Channel width
W_B	Melt pool width

W_C	Equivalent width for kneading discs
W_L	Equivalent width for leakage channel in kneading discs
x	Decision vector
x_i	i – Component of decision vector
X	Decision variable space
z	Objective vector
z_i	i – Component of objective vector
Z	Objective function space

List of Abbreviations

2E	2-Exchange
2S	2-Swap
ACO	Ant Colony Optimization
ACS	Ant Colony System
AS	Adjacent swap
B	Best improvement
B&B	Branch & Bound
CP	Constraint Programming
CPACO	Crowding Population-based Ant Colony Optimisation
CWAC	Component-Wise Acceptance Criterion
EA	Evolutionary Algorithms
EAF	Empirical Attainment Function
F	First improvement
FAN	Fluid Analysis Network
FIFO	First-In-First-Out
GA	Genetic Algorithms
IB	Insert Backward
IF	Insert Forward
ILS	Iterated Local Search
IP	Integer Programming
KB	Kneading block
LH	Left Handed elements
MACS	Multiple Ant Colony System
MOACO	Multi-Objective Ant Colony Optimization
MOAQ	Multi-Objective Ant-Q
MOGA	Multi-Objective Genetic Algorithm
MONACO	Multi-Objective Network ACO
MOOP	Multi-Objective Optimization Problem

MO-PACO	Multi-Objective Population-based Ant Colony Optimization
NSGA	Nondominated Sorting Genetic Algorithm
PAES	Pareto Archived Evolution Strategy
P-ACO	Pareto Ant Colony Optimization
PDM	Particle Dispersed Melting
PED	Plastic Energy Dissipation
PLS	Pareto Local Search
PP	Polypropylene
RH	Right Handed elements
RPSGA	Reduced Pareto Set Genetic Algorithm
SA	Simulated Annealing
SAC	Scalarized Acceptance Criterion
SEMO	Simple Evolutionary Multi-objective Optimizer
SME	Specific Mechanical Energy
SPEA	Strength Pareto Evolutionary Algorithm
TS	Tabu Search
TSCP	Twin Screw Configuration Problem
TPLS	Two-Phase Local Search
VEGA	Vector Evaluated Genetic Algorithm

Chapter 1

Introduction

Intermeshing self-wiping co-rotating twin screw extruders are widely used for compounding operations due to their modular construction, good mixing performance and accurate control of operating parameters. Their geometry can be easily adapted to work with different polymeric systems, e.g., immiscible and reactive polymer blends, composites, nanocomposites...

Unlike conventional single screw extruders, in these machines the screws are built by connecting available elements with different geometrical features along a shaft. Thus, the performance of this type of extruder depends both on the operating conditions (screw speed, barrel temperature and feed rate), and screw configuration. Consequently, defining the best screw configuration to be used in a specific compounding/reactive extrusion operation is a task of great practical importance. The definition of the best screw configuration is a combinatorial problem since it is necessary to identify the location of the available screw elements, being for that reason a challenging and difficult task.

The co-rotating twin screw configuration problem (TSCP) consists in defining the best sequence and location of a pre-defined available set of screw elements (including conveying, reverse and kneading elements) along the screw shaft in order to optimize the process. Due to the huge search space and multi-objective nature of this problem, where several, often conflicting, criteria must be satisfied simultaneously, this task is mostly carried out based on available empirical knowledge, or using modeling software and trial-and-error approaches. Recently, an automatic optimization procedure has been developed to solve the TSCP by coupling an optimization algorithm to a modeling routine. However, this approach is only useful if the modeling program is able to properly predict the

evolution of the main flow parameters (e.g. melt pressure, temperature, residence time, viscosity, shear rate, power consumption) along the screw geometry and is sensitive to changes of the major variables, namely screw geometry, operating conditions and polymer properties.

The global objective of this work is to develop a multi-objective algorithm able to define the best axial location along the screw shaft of a pre-defined number of available screw elements, in order to maximize the process performances, i.e., to obtain a good approximation of the real Pareto front for the TSCP in a reasonable computational time.

This thesis is organized as follows:

Chapter 2 reviews the current state-of-the-art of two topics that are relevant to the research undertaken: i) modeling and optimization of twin screw extrusion; ii) multi-objective strategies and performance assessment tools.

The modeling process of twin screw extrusion is described in Chapter 3, which starts by a description of the physical models considered and the geometric simplifications applied. Then, the computational implementation is described and the accuracy and sensitivity of the program to the variables that influence the extrusion process are tested and the results compared with experimental data.

In Chapter 4 an efficient iterative improvement algorithm is developed and the influence of several neighborhood structures, neighborhood search strategies and neighborhood restrictions are tested for single objective TSCPs. These initial tests allowed the development of an efficient Two Phase Local Search (TPLS) algorithm which is applied to bi-objective TSCPs. The performance of this algorithm is compared to the Reduced Pareto Set Genetic Algorithm (RPSGA), a multi-objective evolutionary algorithm, making use of the Empirical Attainment Function (EAF) comparison methodology.

In Chapter 5 the efficient iterative improvement algorithm developed in previous chapter is extended to the Pareto Local Search algorithm (PLS). The algorithm framework is presented and different experiments are conducted in order to compare its performance with RPSGA and TPLS algorithms.

Chapter 6 discusses the application of a meta-heuristic that takes inspiration on real ants' behavior, the Multi-Objective Ant Colony Optimization algorithm (MOACO). Different algorithm parameters are tested in order to develop an efficient MOACO able to deal with the specificities of the twin screw

configuration problem. Then, the MOACO performance is compared with TPLS and RPSGA algorithms through the application to bi-objective TSCPs using EAFs.

Based on the algorithms considered above (RPSGA, TPLS, PLS and MOACO), in Chapter 7 different hybrid algorithms are proposed. The different possibilities of hybridization are presented and described. Then, their ability to deal with the bi-objective twin screw configuration problems is compared with RPSGA, TPLS, PLS and MOACO algorithms.

Chapter 8 provides a practical application of the automatic optimization procedure by using the RPSGA algorithm for the optimization of starch cationization, a reactive extrusion operation. The model to simulate the reaction is initially presented. Then, several case studies, considering screws with different lengths and screw elements, are discussed.

Finally, in Chapter 9 the major conclusions of this work are stated and proposals for future work are suggested.

Chapter 2

Twin Screw Extrusion and Multi-objective Optimization

2.1. Co-Rotating Twin Screw Extrusion

2.1.1. Introduction

The oldest and simplest extruder used in the polymer industry is the single screw extruder [AGA91]. It comprises one Archimedes-type screw rotating inside a heated barrel and it is widely used in most extrusion lines. However, since this machine presents some limitations, including limited mixing capability, poor control of residence time distribution and interdependence between output and die resistance, a number of alternative designs has been proposed. The most popular are twin screw extruders, which can be offered in a number of designs, depending on the direction of screw rotation and degree of intermeshing (Figure 2.1).

This work focus on intermeshing co-rotating twin screw extruders, which have been extensively used in the polymer industry for compounding operations, but also in other areas such as powder coating, food processing (cereals, snacks, pet food...), chemical, rubber, aluminum and pharmaceutical industries [WHI10]. The first fully intermeshing co-rotating twin screw extruder was proposed and patented by R.W. Easton in 1923 [EAS23]. Later, Erdmenger [ERD64a] patented a modular intermeshing co-rotating twin screw extruder containing both simple screws and kneading disc elements. A detailed historical evolution of twin screw extruders can be found in [WHI10].

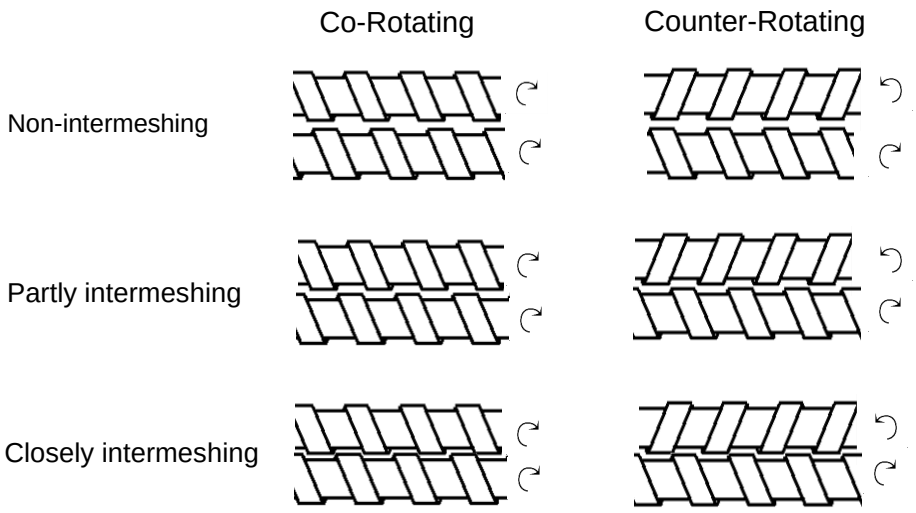

Figure 2.1. Classification of twin screw extruders according to the direction of screws rotation and the degree of intermeshing.

Intermeshing co-rotating twin screw extruders are composed of two parallel and geometrically similar Archimedes-type screws, interlocked as much as possible and rotating at the same speed and same direction, inside a heated barrel.

Screws are usually built up by coupling individual screw elements with different geometrical features. Conveying, mixing (e.g. polygon, turbine mixing, barrel valves, mixing tips, etc) and kneading elements with different geometries (pitch, stagger angle, length, number of starts) are available in the market. Therefore, these machines are versatile and adaptable to the particular requirements of each process, namely in terms of conveying, distributive and/or dispersive mixing, devolatilization or pressure generation.

Co-rotating twin screw extruders usually operate under starved feed conditions, the screws having a sequence of fully filled and partially filled regions. As observed by Erdmenger, the material follows a figure of 8 pattern [ERD64b]. They can generate good dispersive and distributive mixing capabilities and reach high outputs. Melting usually develops quickly and in a short axial length.

Due to the complexity of local geometries (mainly in kneading disc sections), particularly the existence of an intermeshing zone, the development of flow in channels with varying degree of fill, the development of a complex melting process and the significant changes in the material properties along the screw (for example, during reactive extrusion or polymer reinforcement its viscosity level can change various orders of magnitude) the plasticating process is very complex and, consequently, difficult to understand, model, control and optimize.

2.1.2. Modeling

In the last decades considerable attention has been devoted to understanding the phenomena developing in co-rotating twin screw extruders, from the point where the polymer is fed into the hopper until it exits the die. Experiments were carried out, mathematical models were proposed and commercial software is now available on the market.

Different experiment approaches were adopted. One of them is to use the traditional Maddock screw extraction technique developed for single screw extruders [MAD59], in which after the stationary regime has been reached the screws and the polymer are pulled out of the extruder [BAW95, POT96a, VER01]. The second experimental method consisted in the use of a clam shell barrel, as proposed by Todd [TOD93] and Gogos [GOG98], which allowed the barrel to open in few minutes. More recently, a new method has been suggested consisting in flow visualization through glass windows placed in the barrel, or through a complete transparent barrel [SAK95, WON97, CAR99, WON00, LIU01, CAR02, YIC03].

Experiments were carried out in order to study either the entire plasticating process [WON97] or a particular stage, such as solids conveying [BAW95, WON00] or melting [TOD93, BAW95, POT96a, GOG98, VER01, LIU01, YIC03, JUN03], or the behavior of flow in some specific screw elements, such as kneading blocks [CAR99, CAR02].

The knowledge acquired on the phenomenological aspects of the process progressively enabled its mathematical modeling. Some efforts focused on a specific plasticating step, including solids conveying [CAR93, BAW95, POT96b, WON00], melting [POT94, POT96a, BAW98, GOG98, ZHU01, VER01, LIU01, YIC03, JUN08] and melt flow considering just conveying elements [BOO80, DEN80, SZY87a, CHI96, GOF96, CHE98, STR00], kneading discs elements [WER79, SZY87b, SZY88a, SZY88b, GOT90, YAN92, VAN96, CHE97, YOS00, ISH00, BRA00, YOS01, VAL09] or considering a sequence of both conveying and kneading discs [WHI87, MEI88, WAN89b, CHE94a, CHE94b, POT94, KIM00, WHI01a]. Other authors considered the global process, by linking the entire sequence of events, from hopper to die [BAW97, VER98, POT99, CAN99, WHI01b].

Process modeling requires the detailed and precise geometrical description of the co-rotating twin screws which was provided by Booy [BOO78]. Nevertheless, in most cases the semi-circular channel cross-section is approximated to an equivalent rectangular channel [BOO78, WHI10].

The first effort to model solids conveying dates back to 1993 when Carrot et al. [CAR93] developed a mathematical model (similar to the Tadmor model for single screw extruders [TAD72]) for

conveying in the channels and in the intermeshing zone. Bawiskar and White [BAW95] and Potente et al. [POT96b] took into consideration a partially filled feed zone and a completely filled compression zone. Wong et al. [WON00] assumed that the axial solids transportation takes place in the upper and lower intermeshing zones.

Significant experimental work demonstrated that the melting mechanism in co-rotating twin screw extruders occurs in a very short axial length and is quite different from that in single screw extruders [TOD93, POT96a, BAW95, BAW97, GOG98, VER01]. The first melting models proposed were also inspired by the classic model for single screw extrusion developed by Tadmor and Klein [TAD70]. Potente et al. [POT94] and Bawiskar and White [BAW98] assumed the formation of a melt layer contiguous to the hot barrel and a pellet bed at the screws surface. Potente et al. [POT94] took into account melting in kneading blocks that were approximated to a screw of equivalent pitch. Later, Potente and Melisch [POT96a] proposed a melting model based on the existence of individual particles that are uniformly dispersed in the polymer melt. Melting is only considered when the solid particles are wetted with melt. Later, Zhu et al. [ZHU01] suggested a Particle Dispersed Melting (PDM) mechanism simplified into a single-particle-dispersed domain. Gogos and co-workers [GOG98, QIA00, QIA03] demonstrated the importance of Plastic Energy Dissipation (PED) in the heating/melting of solid polymer, whereby individual pellets suffer an irreversible deformation and heating due to repeated compaction and movement. Vergnes et al. [VER01] demonstrated that melting develops in two steps. Initially, a thin film near to the barrel surface is formed, as reported by Bawiskar and White [BAW98]. Then, the solid polymer becomes surrounded by molten polymer. Vergnes proposed a melting model for this second stage [VER01].

Based on the same observations [WON00], Liu et al. also presented a melting model for partially filled screw channel [LIU01]. Different flow patterns in the melting section were described by Yichong and Fuhua [YIC03]. Jung and White [JUN03] carried out several experiments and identified three different melting initiation regimes: melting initiated from the barrel, melting initiated from the screw and melting initiated in the bulk. Accordingly, they put forward mathematical models to predict how melting would develop under different prescribed conditions [JUN08].

Modeling of melt conveying has a longer history than modeling of solids conveying and melting. In 1980, Booy [BOO80] and Denson [DEN80] independently proposed models to take into account the melt flow through individual screw elements by applying analytical and numerical models. Later, a 2-D approach was proposed by Szydlowsky et al. [SZY87a] for a Newtonian fluid based on the

Fluid Analysis Network (FAN) methodology initially proposed by Tadmor et al. [TAD74]. The first approach to consider non-Newtonian fluids was suggested by Wang and White [WAN89a], where the FAN and 3-D Finite Element Methods were applied. Recently, different descriptions of the 3-D flow in conveying screw elements were proposed [CHI96, GOF96, CHE98, STR00].

Flow in kneading blocks is much more complex than in conveying elements. Therefore, the several approaches to model flow in this type of screw elements consider only the peripheral flow in θ-direction around a disc [VER98], or that the total flow rate is the sum of a main flow rate in a continuous equivalent screw channel and a transverse leakage flow between the tips [DEL93]. The flow and mixing behaviors in kneading blocks were first analyzed and modeled by Werner and Eise, by applying a 1-dimensional model [WER79]. Szydlowsky et al. extended the 2-D FAN methodology to the modeling of the flow in kneading discs [SZY87b], the effects of the intermeshing section being included latter [SZY88a]. They also applied the same methodology to non-Newtonian fluids [SZY88b]. Several 3-D flow analyses of kneading discs have been reported in the literature [GOT90, YAN92, VAN96, CHE97, YOS00, ISH00, BRA00, YOS01, VAL09]. Some use complex and advanced numerical techniques that are applied to study the mixing efficiency of this type of screw elements [CHE97, YOS00, YOS01, VAL09, FAR10].

Models considering either Newtonian or non-Newtonian flow through a sequence of conveying and kneading disc elements in a co-rotating twin screw extruder were also proposed [WHI87, WAN89b], some of them including non-isothermal behavior [MEI88, CHE94a, CHE94b, POT94, KIM00, WHI01a]. The first attempt to model the entire plasticating process by combining solids conveying, melting and melt conveying was made by Cheng and White [CHE94b], which was latter integrated in a global simulation program named Akro-Co-Twin-Screw® [WHI01b]. In 1998, Vergnes et al. [VER98] developed a global 1-D simulation software named Ludovic® (Armines) which can quickly and easily run on a personal computer. Its predictions were validated by comparison with experimental data [CAR00]. The initial versions of the software did not consider the melting phase, but the particle dispersed melting mechanism was subsequently integrated [VER01]. Based on analytic flow and heat transfer models (1-D approach), Canedo developed the Twin-Screw Extruder Simulator (TXS™) from Polytech [CAN99]. Similarly, Potente and co-workers [POT99] applying 1-D or 1.5-D approaches developed a composite simulation program (SIGMA, University of Paderborn). More recently, Wilczynsky et al. [WIL12] proposed a new generalized approach for modeling this plastics processing technique.

Mathematical models considering reactive extrusion in co-rotating twin screw extruders have also been developed using either 1-D, 2-D or 3-D flow approaches [POU01a, VER04, CHO04]. A wide range of chemical reaction systems were modeled, such as the anionic polymerization of caprolactam [KYE96], ethylene-vinyl acetate copolymer transesterification [BER98], free radical PP degradation [STR00], ε-caprolactone polymerization [POU01b, ZHU05], methyl methacrylate reactive extrusion [ZAG05], starch cationization [BER07a], peroxide-controlled degradation of polypropylene [BER00, VER00, BER06] and vinyl silane grafting onto PE [FUK00].

2.1.3. Optimization

In industrial practice, the performance of twin screw extruders is dictated by the proper choice of the screw geometry and operating conditions (screw speed, output and barrel temperature). This task is usually accomplished based on available empirical knowledge and/or trial-and-error approaches, without a guarantee that an optimal solution was found. Several sets of operating conditions and screw geometries are tested either using modeling programs or experimentation, until the desired result is achieved. To perform this job a large number of experiments should be performed, which is costly both in terms of time and money.

A few scientific methodologies have been proposed to perform this task:

i) solve the inverse problem, i.e., solve the governing equations of the process in order to the operating conditions and/or geometric parameters (variables to optimize) [COV95];

ii) undertake a partial optimization of the problem, considering separately the several plasticating steps and solving the respective inverse problem [RAU86];

iii) establish a specific design methodology and apply the modeling equations, one per plasticating step [CHU98];

iv) develop a global optimization procedure, by coupling an optimization algorithm to a modeling routine [COV99].

This last automatic optimization procedure was successfully applied to solve several polymer engineering problems [GAS00, GAS04, FER10, FER11]. Figure 2.2 shows the optimization methodology proposed. First, the user defines the optimization problem, comprising the definition of the type of optimization (maximization or minimization problem), the parameters to optimize and the objectives. Then, initial parameters values (e.g. operating conditions and/or screw

geometry/configuration) are proposed by the optimization algorithm (initial solutions). Taking into account these specific values, the modeling package predicts the extruder's response by evaluating the proposed solutions. The performance of these solutions is used (in terms of objectives) by the optimization algorithm to determine new better solutions for the subsequent iterations. This process is repeated until a stop criterion is reached. At the end, the best solutions obtained along the search process are presented to the user.

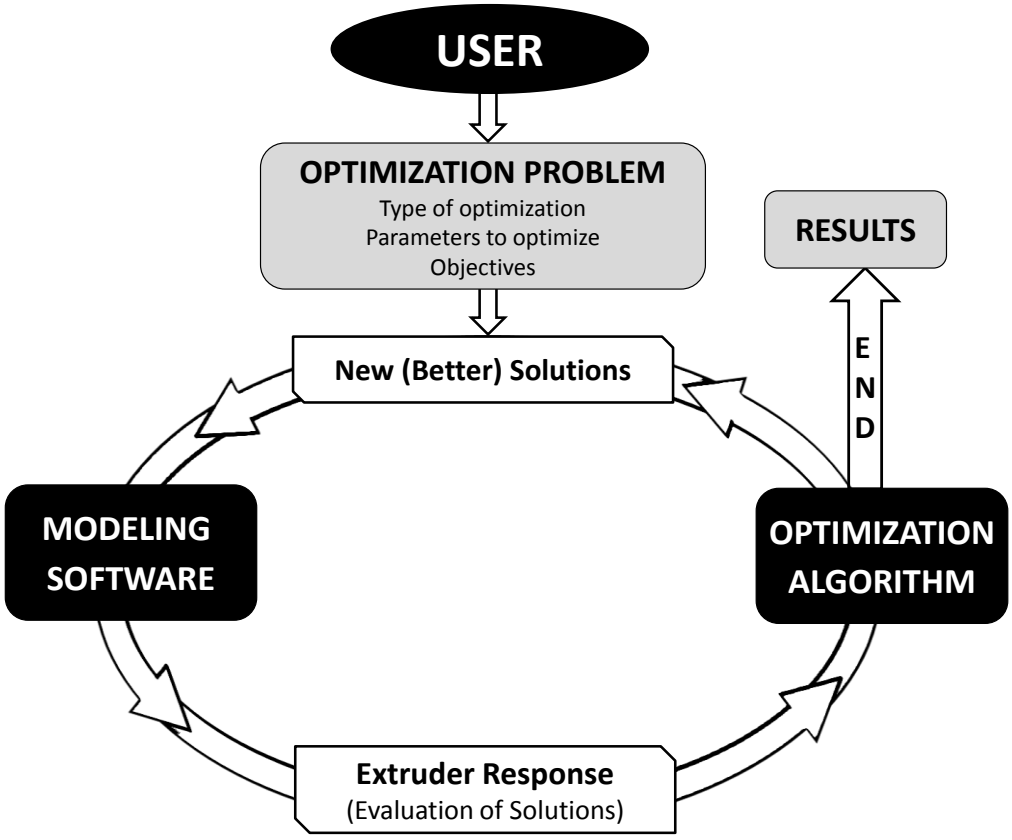

Figure 2.2. Optimization methodology.

This methodology was implemented to define the operating conditions and/or screw geometry of single screw extruders [COV99, GAS00, GAS01, GAS06]. Concerning co-rotating twin screw extruders, only a limited number of studies can be found in the open literature. They deal with the definition of the best operating conditions [GAS02] and/or screw geometry/configuration for simple compounding processes and for ε-caprolactone polymerization [GAS05, POT06].

If the problem of setting the best operating conditions of a modular co-rotating twin screw extruder is not an easy task, that of defining the best screw geometry/configuration is much more complex.

It usually consists of:

a) defining the location along the screw axis of a fixed set of screw elements, the aim being to maximize the process performance;

b) selecting the location along the screw axis of a given number of individual screw elements (conveying, reverse and kneading discs elements) from a wider available set.

Another possibility is to design and/or improve the geometry of a specific screw element, for example, to define the best pitch and/or length of a conveying screw element, as done by Potente et al. [POT06]. In this case, the variables to optimize are continuous.

2.2. Multi-Objective Optimization

2.2.1. Definitions

The purpose of an optimization problem is to choose the best solution from a set of available alternatives, i.e., the solution with the minimum or maximum value of the objective function for minimization or maximization problems, respectively.

Real world problems often involve the simultaneous optimization of various, often conflicting, objectives. These problems are identified as multi-objective optimization problems, the aim being to define the best set of decision variables that simultaneously satisfies all constrains and optimizes a vector of objective functions.

Definition 2.1. Formally, a *multi-objective optimization problem* (MOOP) is defined as:

$$\text{maximize } f(x) = \left(f_1(x), ..., f_k(x) \right)$$

$$\text{subject to } x \in X$$

where $x = (x_1, x_2, ..., x_m) \in \Re^m$ is the *decision vector* composed by m decision variables, $X \subset \Re^m$ the *feasible set* defined by equality and inequality constraints and $f : X \to Z$, $f = (f_1, f_2, ..., f_k) \in \Re^k$ the *objective function* that evaluates the quality of a specific solution. This *k*-dimensional function assigns an *objective vector* $z = \left(f_1(x), f_2(x), ..., f_k(x) \right)$ in the set $Z = f\left(X\right)$ known as *feasible set in the objective function space*. The sets $\Re^m$ and $\Re^k$ are denoted

as *decision variable* and *objective function spaces*, respectively. The search spaces are represented in Figure 2.3.

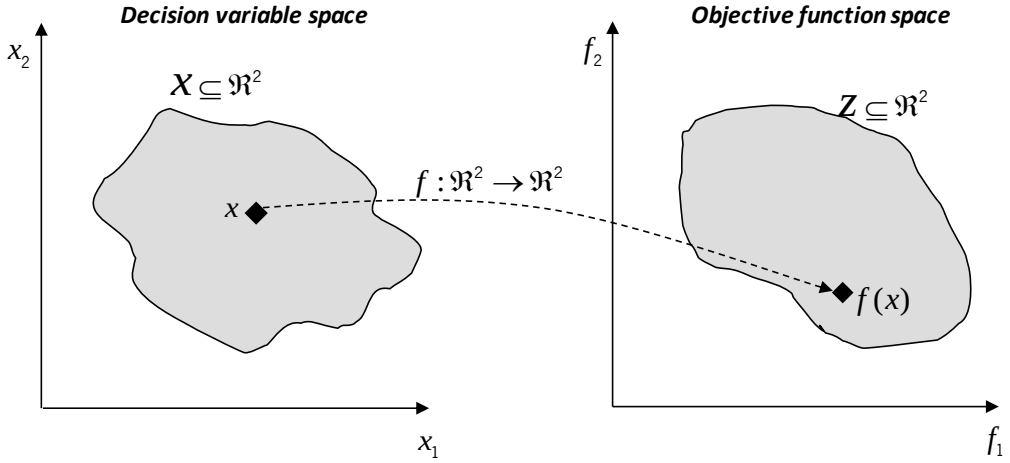

Figure 2.3. Decision variable space and objective function space.

In single objective optimization problems the candidate solutions can be sorted unambiguously. Assuming a maximization problem, a solution a is better than b if and only if $g(a) > g(b)$, with $a,\ b,\ g(a)$, and $g(b) \in \Re$ and $g : \Re \to \Re$ the objective function. With this simple relation of order it is possible to sort all solutions, the optimal solution being easily identified. In multi-objective optimization, where several conflicting objectives must be optimized simultaneously, this is usually not possible. Note that an improvement in one objective can only be obtained at the expenses of the degradation of at least another objective. In this case, as there is no canonical order to allow ranking all solutions from the worst to the best and there is no information about the decision maker preferences, a different relation of order must be properly defined. Among the different relations of order existing in literature, the most popular is the *Pareto dominance relation* that was first proposed in 1881 by Francis Edgeworth [EDG81] and generalized by Vilfredo Pareto [PAR96]. Without loss of generality, a maximization problem will be assumed throughout the remaining of the chapter. Note that similar definitions can be considered for the minimization case.

Definition 2.2. Relations of order

Consider the vectors $u = (u_1, ..., u_k)$ and $v = (v_1, ..., v_k) \in \Re^k$. The following orders are defined:

weak component-wise order as $u \geq v$ *iff* $\forall i \in \{1, 2, ..., k\}\ u_i \geq v_i$

component-wise order as $u > v$ *iff* $\forall i \in \{1, 2, ..., k\}\ u_i \geq v_i \ \wedge\ \exists i \in \{1, 2, ..., k\} : u_i > v_i$

Definition 2.3. Pareto dominance

For any two vectors $u=(u_1,...,u_k)$ and $v=(v_1,...,v_k) \in \Re^k$ is possible to say that:

u *dominates* v, denoted as $u \succ v$, $\quad iff \quad u > v$

u *weakly dominates* v, denoted as $u \succeq v$, $\quad iff \quad u \geq v$

u *is incomparable or non-dominated to* v, denoted as $u \nsucc v \quad iff \quad u \not\geq v \ \wedge \ v \not\geq u$

An example of Pareto dominance relation in $\Re^2$ is illustrated in Figure 2.4. As z_1 has a higher value of f_2 when compared with z_3 and the same value of f_1, z_1 dominates z_3 $(z_1 \succ z_3)$. Also, z_2 dominates z_3 given that the value of the first objective of z_2 is higher than z_3, for the same value of f_2 $(z_2 \succ z_3)$. However, z_1 and z_2 are non-dominated or incomparable: z_2 is better than z_1 with respect to f_1 but z_1 is better than z_2 with respect to f_2. In this case, $z_1 \nsucc z_2$.

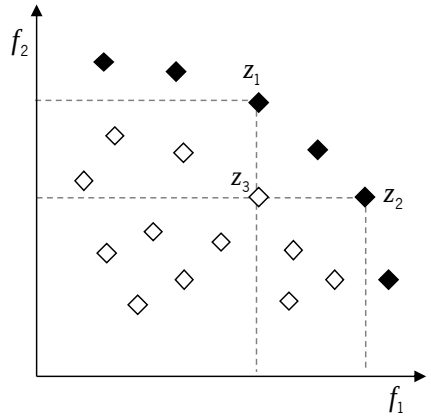

Figure 2.4. Pareto Dominance Relation in $\Re^2$.

Definition 2.4. Pareto Optimal

A solution $x \in X$ is said to be *Pareto Optimal* with respect to X if and only if there is no $x' \in X$ for which $f(x') = (f_1(x'),..., f_k(x'))$ dominates $f(x) = (f_1(x),..., f_k(x))$.

Definition 2.5. Pareto Optimal Set

The *Pareto Optimal Set*, *P**, is the subset $X^* \subseteq X$ that contains *only* and *all* Pareto optimal solutions. Formally, it is defined as:

$$P^* = \{x \in X : \neg \exists x' \in X \ f(x') \succ f(x)\}$$

Definition 2.6. Pareto Front

The image of the Pareto Optimal Set *P** in objective space is called *Pareto Front FP** and is defined as:

$$FP^* = \{ f(x) = (f_1(x), ..., f_k(x)) : x \in P^* \}$$

The concepts of Pareto Optimal Set and Pareto Front are illustrated in Figure 2.5. The closed points represent the Pareto optimal decision vectors (in the decision variable space) and the Pareto optimal objective vectors (in the objective function space). Open points represent decision vectors (in the decision variable space) which are dominated (in the objective function space) by any closed point. The set of Pareto optimal decision vectors is referred to as the Pareto Optimal Set and the set of Pareto optimal objective vectors as the Pareto Front. Note that the Pareto Front only comprises non-dominated solutions.

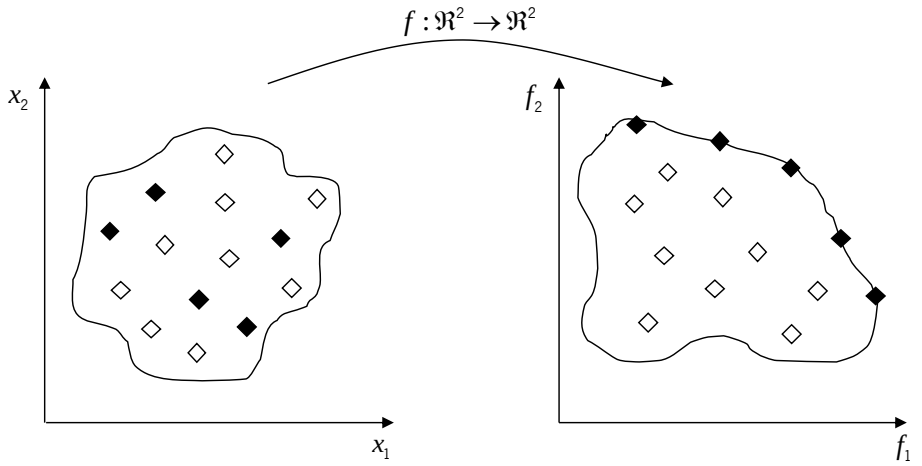

Figure 2.5. Pareto Optimal Set and Pareto Front.

Simultaneously with the search process, solving a multi-optimization problem also involves decision making, since only a single solution must be chosen to take into account the preferences of the decision maker. Decision making can be made:

i) *a priori*, by scalarizing the objectives into a single objective function transforming the problem in a single optimization problem. In this case, each objective is pre-multiplied by a weight defined *a priori* by the decision maker, which qualifies the relative importance of each objective. However, if the preferences of the decision maker are changed, a new optimization procedure must be carried out;

ii) *iterative* (during the search), by incorporating the preferences of the decision maker at each optimization step;

iii) *a posteriori*, by optimizing simultaneously all the objectives and at the end selecting the solution that best satisfies the needs of the decision maker [MIE99].

If no information about the preferences of the decision maker can be incorporated before or during the search process, none of the Pareto-optimal solutions can be considered better than the remaining. In this case, the problem must be tackled in terms of Pareto concepts, the aim being to find the solutions that are not worse than any other solution and better in at least one of the objectives. In other words, the search process must provide the Pareto Optimal Set or a set of non-dominated solutions. The role of the decision maker is to select the solution that best meets his/her needs. Thus, a good optimizer is an algorithm that is able to provide the Pareto Optimal Set or a good approximation to this set in a reasonable computation time.

2.2.2. Heuristics

The role of multi-objective optimization is the determination of the entire Pareto Optimal Set, i.e., the non-dominated set in the entire feasible search space. Exact algorithms are able to perform this task, but for small instances and requiring lot of time. Simultaneously, they prove to be inefficient and incapable to deal with optimization problems when the complexity and size grows, mainly due to the exponential increase in computational time.

Given the features of real world optimization problems (great complexity and large size), it seems unfeasible to design efficient exact algorithms. Thus, in these situations it is desirable the use of algorithms that provide a good approximation to the Pareto Optimal Set in a reasonable computation time, even without guarantying optimality. These algorithms are called heuristics.

In this section a brief description of basic local search algorithms is initially presented. Then, more complex heuristics, known as metaheuristics, are introduced and briefly described.

A. Basic Local Search: Iterative Improvement

Iterative improvement algorithms are based on the concept of neighborhood search and are considered basic local search procedures. The algorithm starts from a random initial solution x and scans its neighborhood (which is previously defined according to the problem under study) searching for a better solution, i.e., a solution which improves the value of the objective function [HO004]. Then, this better solution will be defined as the current solution and the search process continues. The movement from the current solution to a neighbor solution can be made in two different ways: *first improvement*, in this case the algorithm moves to the first better solution found; *best improvement*, the algorithm first checks the entire neighborhood and then moves to the best solution found [PAP82]. In both cases the algorithm stops when no improvement is obtained. The solution found is optimal, but there is not guarantee that it is the global optimum. This constitutes the big disadvantage of iterative improvement algorithms, i.e., the incapacity to continue the search process when trapped in a local optimum.

The performance of this type of methods is strongly dependent of an appropriate neighborhood structure, which should be carefully defined considering the specificities of the multi-objective problem under study [HO004].

B. Metaheuristics

In recent years, researchers have increased their interest in a new class of approximate algorithms called metaheuristics. This interest is due to the ability of the metaheuristics to provide very good solutions in a reasonable computation time. Another attractive characteristic is the possibility that they have to be easily adapted to a wide set of problems with few modifications and without making structural changes in the algorithms, i.e., they are not problem-specific. They also have the capability to get away from the local optimum and continue the search process in other regions through the application of different techniques, as will be described below. Metaheuristics are characterized by making use of exploration and exploitation search processes. Exploration is used to diversify the search process along the entire search space and exploitation is used to intensify the search in promising search areas.

Unlike the iterative improvement algorithms, they do not have an intrinsic stopping criterion. Instead, a pre-defined criterion set by the user, like the number of iterations or computational time, is used [VOß09].

Numerous definitions have been proposed in order to properly define and characterize metaheuristics. For a better understanding of metaheuristics, the following two definitions are transcribed here:

> "Metaheuristics are typically high-level strategies which guide an underlying, more problem specific heuristic to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descent by allowing the local search to escape from local optima. This is achieved by either allowing worse worsening moves or generating new starting solutions for the local search in a more intelligent way than just providing random initial solutions. Many of the methods can be interpreted as introducing a bias such that high quality solutions are produced quickly. This bias can be of various forms and can be cast as descent bias (based on objective function), memory bias (based on previously made decisions) or experience bias (based on prior performance). Many of the metaheuristics approaches rely on probabilistic decision made during the search. But, the main difference to pure random search is that in metaheuristic algorithms randomness is not used blindly but in a intelligent, biased form." [STU98a]

> "A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method. The family of metaheuristics includes, but is not limited to, adaptive memory procedures, tabu search, ant systems, greedy randomized adaptive search, variable neighborhood search, evolutionary methods, genetic algorithms, scatter search, neural networks, simulated annealing, and their hybrids." [VOß01].

The diversity of metaheuristics can be classified considering various characteristics [BLU03]:

i) *Nature-inspired vs. non-nature inspired*, i.e., considering the origin of the metaheuristic. Some are inspired in nature, such as Genetic Algorithms (GA) and Ant Colony Optimization (ACO) algorithms, while others, such as Tabu Search (TS) and Iterated Local Search (ILS), are considered non-nature inspired.

ii) *Population-based vs. single point search*, i.e., taking into account the number of solutions considered simultaneously. Population-based algorithms perform the search process using a set of solutions. After each iteration a set of solutions is returned and will be the starting set for the next iteration. In single point search algorithms only one solution is considered.

iii) *Dynamic vs. static objective function,* i.e., considering how the objective function is used. The same objective function can be used during the entire search process (static objective function) or the objective function can be changed, for example so that the algorithm moves from a local optimum (dynamic objective function).

iv) *One vs. various neighborhood structures*, i.e., taking into account the number of neighborhood structures used on the search process.

v) *Memory usage vs. memory-less methods*, i.e., considering the use of memory along the search process. In some metaheuristics (memory-less methods) memory is only used to decide the next step on the search process, avoiding cycles. In other cases, memory is used in the long-term (memory usage), which helps the diversification and intensification of the search process.

In this work, the *population-based vs. single point search* classification will be considered. In the next sections the basic concepts of some of the most common trajectory and population-based methods used in literature are briefly described. As examples of trajectory methods the Iterated Local Search (ILS), Simulated Annealing (SA) and Tabu Search (TS) algorithms are presented. In the case of population-based methods, Evolutionary Algorithms (EA) and Ant Colony Optimization algorithm (ACO) are taken into account.

Trajectory algorithms (single point search)

Iterated Local Search. The Iterated Local Search algorithm (ILS) is a simple but powerful metaheuristic proposed by Lourenço et al. [LOU02]. The idea is to apply a perturbation to the optimum local solution in order to proceed with the search process and find a solution as close as possible to the global optimum. The algorithm starts from an initial solution x randomly generated which will be improved by a local search procedure until a local optimum x' is found. Then, a perturbation is applied to x' generating a new solution and a local search procedure is employed until a new local optimum x'' is reached. Finally, an acceptance criterion is considered to compare the two local optima solutions x' and x'' and choose the one that will be used to proceed with the search.

Particular attention must be paid to the perturbation mechanism, since it can affect the effectiveness of the algorithm. A too small perturbation may not allow the algorithm to escape from the attraction of the local optimum. In the case of a larger perturbation, the algorithm can become a random search process.

Simulated Annealing. Simulated Annealing (SA) is a stochastic method based on the neighborhood search and is inspired by the annealing process of metals [KIR83, CER85]. When a metal is heated above its melting temperature and then cooled slowly, it reaches a configuration with lower internal energy than the initial one – thermodynamic equilibrium. However, if the cooling process is not sufficiently slow, the final configuration of the material is not a perfect crystal revealing irregularities and imperfections.

The analogy between the annealing process and optimization is made as follows: the feasible solutions are the solid states, the value of the objective function is the energy of the diverse states, the optimum solution is the state of minimum energy and the control parameter is the temperature. Therefore, the simulated annealing algorithm involves a sequence of iterations that change the current solution x accepting the modifications that improve the objective value. However, to avoid premature convergence to a local optimum, the algorithm tolerates moves from the current solution x to a neighbor solution x' without improvement of the objective function by incorporating a probabilistic acceptance criterion. The most used is the Metropolis acceptance criterion [MET53]. Accordingly to this criterion, a movement from the current solution x to the solution x' will be accepted if some improvement of the objective function value is attained. Otherwise, it will be

accepted with a certain probability, that is computed as a function of the quality of the solutions of x and x' and a control parameter T, $e^{|f(x)-f(x')|/T}$.

The probability of accepting worse solutions decreases as the search continues progressively by reducing the value of parameter T. As a consequence, the algorithm gets more selective. An initial high probability allows the algorithm to run away from poor quality of local optima solutions found in the early search stages.

Summarizing, simulated annealing can be considered as a simple extension of iterative improvement algorithm that allows movements to a worse solution accordingly with a probabilistic function.

Multi-objective versions of the Simulated Annealing algorithm were proposed by Serafini [SER94], Czyzak and Jaszkiewicz [CZY97, CZY98], Ulungu et al. [ULU99], Suppapitnarm et al. [SUP00] and Suman [SUM02].

Tabu Search. Another approach to overcome the disadvantage of the iterative improvement algorithm was proposed in 1986 by Fred Glover. It is a metaheuristic based on the concept of flexible memory [GLO86, GLO89, GLO90, GLO97]. Tabu Search (TS) maintains a list of forbidden moves, known as tabu list, and avoids returning to the most recently visited solutions. At each iteration the neighborhood subset (limited to the neighbor solutions that do not belong to the tabu list) of the current solution x are evaluated and the solution with higher improvement is selected. If none of the solutions improves the actual, the solution with smaller deterioration is considered. This new solution becomes the current solution and is added to the tabu list. Since a pre-defined size for the tabu list is defined, one of the existing solutions must be removed, usually using the first-in-first-out concept (FIFO). Thus, the tabu list changes along the successive iterations. Given the dynamic definition of the neighborhood, TS can be classified as a dynamic neighborhood search technique. However, some exceptions to the tabu list acceptance criterion can be considered, for example accepting a tabu movement if it looks attractive. For that purpose, TS uses an aspiration criterion that defines under which conditions a tabu solution can be accepted and released from the tabu list [GLO95]. The most used acceptance criterion accepts a tabu move if it generates a better solution than all the best solutions obtained so far. This avoids the loss of good solutions along the search process just because they are included in the tabu list.

To implement an effective Tabu Search algorithm several parameters must be carefully defined, such as the local search procedure, the neighborhood structure, the tabu conditions, the aspiration conditions and the stopping criterion.

Several approaches to extend the Tabu Search to the multi-objective case have been proposed over the years [HER94, GAN97, HAN97, HAN00, BAY99, BAY01, HO02, JAE04, KUL06, XU06].

Population-based Algorithms

Evolutionary Algorithms. Evolutionary algorithms are based on evolutionary principles [BAC96], comprising evolutionary programming [FOG62, FOG66], evolutionary strategies [REC73], genetic programming [KOZ91] and genetic algorithms [HOL75]. The main differences between the EAs proposed in literature concern the coding representation of the solutions, the selection mechanisms and the structure of the operators. In this work, Genetic Algorithms (GA) will be used and for that reason they will be studied in more detail.

Genetic Algorithms (GA) were proposed by Holland and take inspiration from the natural process of species evolution [HOL75]. According to Darwin's theory, the individuals with the best characteristics are more likely to survive and reproduce, while the less fitted individuals tend to disappear. Consequently, these hereditary characteristics will become more common in the following generations as a result of selection, mutation and reproduction processes, until becoming dominant [DAR59]. This is the natural selection concept: "stronger" individuals have more chances to survive.

By analogy with Genetics, GA deal with a population of solutions called chromosomes that are constituted by units named genes. This population is obtained randomly at the beginning of the search process. At each iteration, the performance of each individual is computed via an objective function (fitness) and some individuals are selected to be part of the reproduction process. Several techniques can be applied to perform the selection, namely roulette-wheel [BAK87], tournament, stochastic universal [BAK87], local [GOR91] and truncation selection [BLI95]. Leaving out the specificities of each technique, selection is generally based on a probabilistic function dependence of the individual fitness, the "best" individuals have a higher probability of being selected and, consequently, of generating a new "better" population. However, this process also considers the

survival of less fit individuals maintaining the diversity of the population, in order to avoid a premature convergence.

Reproduction is based on two main operators: crossover and mutation. The first is applied by interchanging the genes of two parents to obtain two new solutions, hopefully with better characteristics. This allows the algorithm to continue the search in promising regions, and thus obtain more frequently individuals with good genes [GOL89]. Random changes must also be implemented to prevent the algorithm from being confined to some search areas, disregarding new promising areas. This is achieved by applying the mutation operator that selects randomly a specific gene in the chromosome and "changes" its value [BEA93]. The search process finishes when the number of generations (pre-defined parameter) is reached.

The first application of Evolutionary Algorithms to multi-objective problems was proposed by Schaffer [SCH84] with the Vector Evaluated Genetic Algorithm (VEGA). Since then, several algorithms have been proposed, such as, the Multi-Objective Genetic Algorithm (MOGA) [FON93], the Nondominated Sorting Genetic Algorithm (NSGA, NSGA-II) [SRI94, DEB00, DEB02], the Strength Pareto Evolutionary Algorithm (SPEA, SPEA2) [ZIT99, ZIT01], the Pareto Archived Evolution Strategy (PAES) [KNO99, KNO00] and the Reduced Pareto Set Genetic Algorithm (RPSGA) [GAS04].

Ant Colony Optimization. In 1992 Marco Dorigo suggested a population-based algorithm taking inspiration in the foraging behavior of real ants, and named it Ant Colony Optimization (ACO) algorithm [DOR92, DOR99, DOR04]. Ants are social insects that communicate via chemicals by releasing pheromone. When an obstacle appears on the path between the nest and the food source, they are confronted with a dilemma: which side of the path must be chosen in order to take the shortest way? Since they do not have any additional information, in a first phase they will turn to one or the other path with equal probability. While walking along the path, ants will deposit a certain amount of pheromone and as time evolves the amount of pheromone will be higher in the shorter path. Consequently, ants will choose the path with the highest quantity of pheromone.

The ACO algorithm is based on a probabilistic solution construction, where the probabilities are a function of the pheromone strength on the ants' "trail" and of the available heuristic information about the problem. At the beginning, all pheromone trails are initialized with the same pheromone value. Then, each ant will iteratively construct a candidate solution by choosing, at each

construction step, a solution component according with a probability that is proportional to the pheromone value and heuristic information, if available. After a solution is completed it will be evaluated. The components of the best solution are reinforced through the deposition of an amount of pheromone that typically depends on the solutions' quality. The modification of pheromone trails during the search process reflects the collective experience inducing a search around the best solutions found.

The premature convergence of the algorithm is avoided by the application of a (virtual) pheromone evaporation mechanism. The pheromone trails of all the solution components are decreased by a predetermined value, allowing the exploration of new regions of the search space. The algorithm stops when the number of tours (iterations) is reached. The ACO algorithm mimics the cooperation behavior of real ants and the exploration in several directions of the search space by using multiple ants and pheromone trails. The first version of the ACO algorithm (Ant System - AS) employs all the ants that completed the tour to update the pheromone values [DOR92]. Since then, several versions have been proposed, (see [DOR10] for an overview), the most popular and best performing versions being the Ant Colony System [DOR97] and the Max-Min Ant System [STU98b, STU00]. The Ant Colony System (ACS) differs from its predecessor by the use of a local pheromone update (at the end of each construction step by all the ants), the use of only the best ant tour to update the pheromone trails in the end of each iteration (global pheromone update) and the application of a decision rule (pseudo-random proportional rule) that depends on a random variable making a more efficient balance between exploration and exploitation. Similarly, the Max-Min Ant System updates the pheromone trails considering only the best ant of each iteration. It is also characterized by the incorporation of the minimum and maximum pheromone values avoiding stagnation of the search process.

Some ACO algorithms dealing with multi-objective problems were proposed, such as, Multi-objective Ant-Q (MOAQ) [MAR99a], BicriterionAnt [IRE01], Pareto Ant Colony Optimization (P-ACO) [DOE04], Multiple Ant Colony System (MACS) [BAR03], Multi-objective Network ACO (MONACO) [CAR03], SACO [TKI02], COMPETants [DOE03], Multi-Objective Population-based Ant Colony Optimization (MO-PACO) [GUN03] and Crowding Population-based Ant Colony Optimisation (CPACO) [ANG07]. A detailed classification of MOACO algorithms can be found in [ANG09].

Hybrid Metaheuristics

Over the last few years, a recent type of heuristics has drawn the interest of the optimization community: hybrid metaheuristics.

Hybrid metaheuristics do not follow the concepts of a single metaheuristic, but they unite concepts of several algorithmic ideas in order to merge the advantages of different pure algorithms in a synergic way. Several hybrid metaheuristics have been proposed, and proved to be more efficient than "pure" metaheuristics by decreasing the computational effort and/or increasing the quality of the solutions [BLU08]. If in the early days, optimization techniques co-existed without interaction, today the advantage of combining components from different search techniques is commonly accepted.

Several hybrid metaheuristics have been developed, resulting from the combination of components of different metaheuristics, from metaheuristics with problem-specific algorithms, or from other optimization techniques based on operation researchers and artificial intelligence fields (e.g., constraint programming (CP), branch & bound (B&B), integer programming (IP), tree-based search methods, data mining techniques, etc). A review can be found in [TAL02, EHR08, BLU08, BLU10].

A common hybridization methodology incorporates local search procedures into population-based metaheuristics, in order to overcome the drawbacks of slow convergence and random constructions of the latter [ULD91, MUH98, MOS99, DOR04]. This has the advantage of the identification and diversification abilities to find promising areas of the search space provided by population-based metaheuristics as well as the advantage of the local search procedures to intensify the search in promising areas, entailing a more structured exploration.

Metaheuristics provide good solutions in acceptable times, but without any guarantee of the quality of the solution; one the other hand, exact algorithms provide the true Pareto front but require a great amount of computation time. Thus, hybridization of metaheuristics with exact algorithms is also very popular.

Different taxonomies were suggested to provide a common terminology [COT98, TAL02, BLU05, PUC05, RAI06]. C. Cotta [COT98] proposed a classification based on the intensity of the link between the knowledge of the problem and the basic model (strong *vs.* weak). A mixed classification considering either a hierarchical or flat scheme was proposed by Talbi [TAL02]. Puchinger and Raidl [PUC05] proposed a classification that distinguishes between collaborative and integrative combinations. The authors considered the hybridization of metaheuristics with exact

optimization techniques. Blum and Roli [BLU05] classified hybrids based on how algorithms are hybridized: component exchange among metaheuristics, cooperative search and integrating metaheuristics and systematic methods. More recently, Raidl [RAI06] tried to unify the previous taxonomies and presented four criteria to classify hybrid algorithms:

i) *Type of algorithms involved.* Three distinct classes of hybridization can be considered: metaheuristic with metaheuristic, metaheuristics with problem-specific algorithms and metaheuristics with other optimization techniques [BLU05].

ii) *Level of hybridization.* Hybrid algorithms can be defined as high or low level depending on the strength linking of the algorithms. In high level hybridization, a weak coupling between the algorithms exists and each algorithm retains its own identity. Contrarily, in low level combinations there is a strong dependency between the algorithms by exchanging individual components [COT98, TAL02].

iii) *Order of execution*. Each individual algorithm can be executed in a sequential, intertwined or parallel way.

iv) *Control strategy.* If the hybridized algorithms exchange information between then without making part of each other, they are known as collaborative combinations. However, if one of the algorithms is a subordinate component of other, it is classified as integrative [COT98, PUC05].

2.2.3. Performance Assessment Tools

The main objective of this work is the design of an efficient multi-objective algorithm able to deal efficiently with the Twin Screw Configuration Problem (TSCP). For that purpose, several algorithms may be developed and tested using different instances of the problem. Therefore, it is important to employ an appropriate performance assessment methodology to identify which is the best methodology tested.

In single optimization problems, comparing two optimizers is an easy task. A single solution is obtained and its performance can be compared taking into account the solution quality and/or the computation time. If no limit in time is imposed, the performance of the algorithms should be

assessed by a statistical analysis considering both computational time and respective outcome quality.

In the multi-objective case, the performance of two optimizers can also be assessed take into account the outcome and the respective computational time. If a limit in the running time of the algorithm is imposed, the comparison will be only performed taking into account the respective outcomes. Since the result of multi-objective problems is a set of solutions, comparison is not an easy task. As an additional factor, it is necessary to deal with the stochastic nature of the optimizers, i.e., different runs produce a different approximation to the Pareto frontier. Therefore, a statistical analysis must be made to compare the performance of the different optimizers by confronting the corresponding approximations set of samples.

Different performance measures have been conceived. Zitzler et al. [ZIT03, ZIT08] and Knowles et al. [KNO06] reviewed the state of art concerning the statistical performance assessment and discussed the limitations of each of the existing performance measures. Currently, two main approaches are used: quality indicators and attainment functions. The first assign a real value to each approximation set (unary quality indicator); or can assign a value to each set of pairs (binary quality indicator). The most usual unary measure is the hypervolume indicator I_H proposed by Zitzler and Thiele [ZIT99]. It returns the hyper-volume of the objective space that is weakly dominated by an approximation set. The higher the hypervolume value, the better the algorithm. Binary measures are based on the pairwise comparison of sets of non-dominated objective function values. Their main drawback is that they return a pair of values. Moreover, the information returned by these measures usually grows quadratically as a function of the number of runs performed. The main disadvantage of unary and binary measures is the reduction of a multi-dimensional object into one or two scalar values, leaving out many details. On the other hand, they are a good initial indicator of the performance of multi-objective algorithms.

Fonseca and Fleming [FON96] proposed a distinct approach to measure the performance of optimizers, the attainment function method, that has a special statistical meaning. This methodology assigns to each objective vector z in the objective space the probability that z is attained in a single run (i.e., weakly dominated). The true attainment function is not possible to compute, but it can be estimated based on the approximation set samples, i.e., different approximations obtained in several runs. This approximation is denoted as the empirical attainment function (EAF) [GRU01, GRU10]. For example, Figure 2.6a represents the hypothetical outcomes of

three different runs (for a maximization problem) and Figure 2.6b the respective relative frequencies of the distinct regions of the objective space. A point z with a probability of 3/3 means that it is weakly dominated by the three runs performed. Contrarily, a point with a probability of 0/3 means that it is not weakly dominated by any of the runs performed.

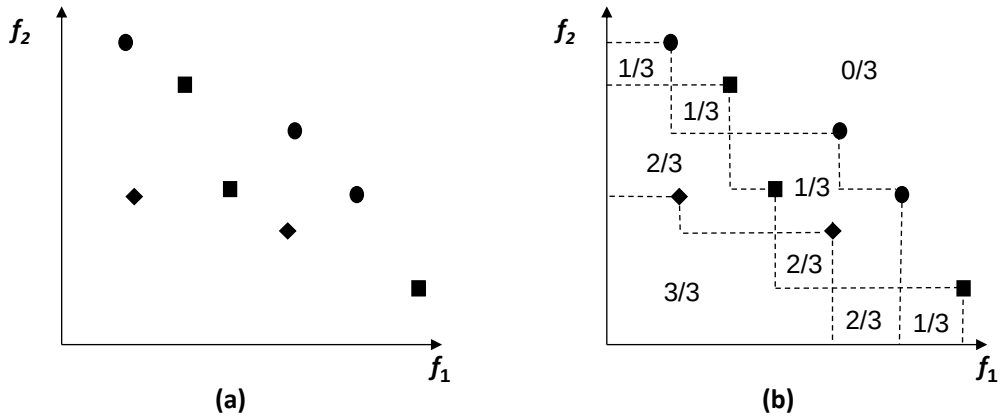

Figure 2.6. (a) Hypothetical outcomes of three runs in a maximization problem; (b) Relative frequencies for the distinct regions in the objective space.

If the comparison between two multi-objective optimizers is carried out by calculating and plotting the differences between the respective EAFs it becomes possible to identify and visualize where and how much the differences between the two optimizers exist and whether they are significant [LOP06, LOP10a]. For that purpose, two distinct steps must be performed. First, the EAFs of each algorithm must be computed. The empirical attainment function is estimated by running each algorithm a certain number of times with a different initial random seed. Then, for each point in the objective space, the differences between the EAFs of the two algorithms are computed and the differences (above 0.2) are plotted.

Figure 2.7 represents the EAFs of two algorithms and Figure 2.8 plots the location of the differences between the respective EAFs. Figure 2.8 left represents the points where the algorithm 1 is significantly better than algorithm 2. The opposite is represented in Figure 2.8 right. The values of the differences are encoded using a grey scale (where darker points represent larger differences between the EAFs) and the worst, median and best results attained by the algorithms are also plotted. In this thesis the plots related with the EAFs were obtained making use of the tools provided by López-Ibanez [LOP10a].

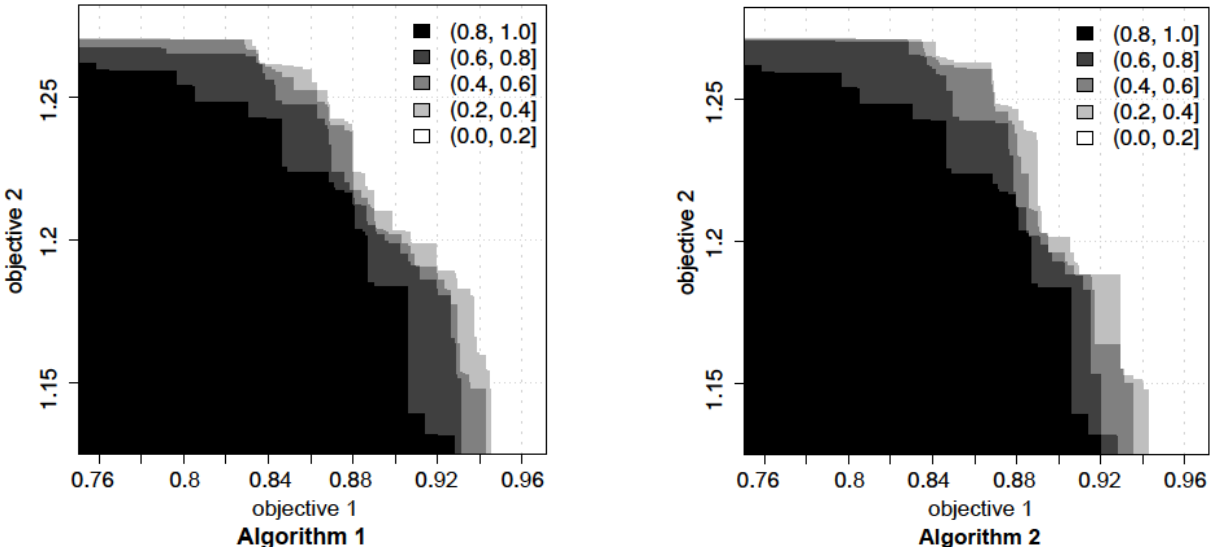

Figure 2.7. Empirical Attainment Functions of two algorithms.

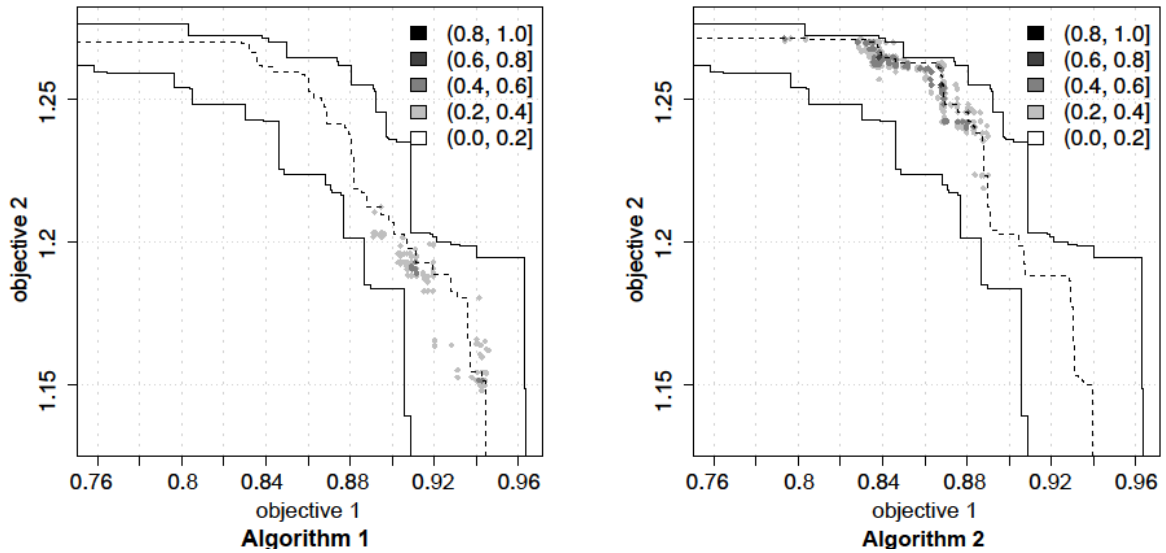

Figure 2.8. Location of differences in terms of EAFs between Algorithm 1 and Algorithm 2. Advantages in favor of Algorithm 1 are indicated on the left side; those in favor of Algorithm 2 on the right side.

2.3. Objectives of the research

Due to the presence of discrete decision variables (location of screw elements along the screw axis) that can be combined with continuous decision variables (geometric characteristics of specific screw elements and/or operating conditions) and the existence of a wide search space (note that the choice of the location of 15 screw elements involves 15!=1307674368000 possibilities), the complete problem of designing an efficient screw profile for modular intermeshing co-rotating twin screw extruders is very complex and challenging.

Figure 2.9 illustrates a TSCP where 15 screw elements (including 10 conveying elements, 3 kneading blocks and 2 left handed elements) must be located along the screw axis.

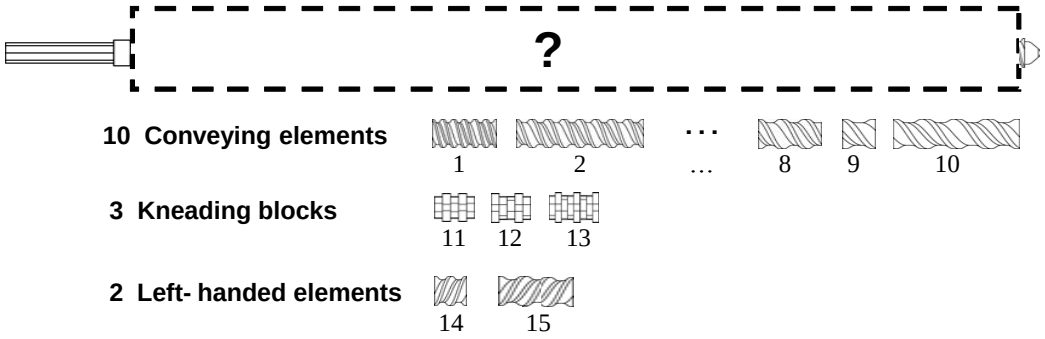

Figure 2.9. Representation of the Twin Screw Configuration Problem.

Therefore, in this work the aim is to define the best screw configuration, i.e., to identify the location of a pre-defined set of screw elements along the screw axis that optimizes the global performance of the operation. This problem is denoted as Twin Screw Configuration Problem (TSCP). It can be seen as a scheduling problem [HOO04], where the aim is to define the best order of a given operation from a set of operations that are required to be accomplished.

Considering the global aim of the research, the following objectives are sought:

i. To develop an appropriate modeling routine able to predict the evolution of flow and heat transfer parameters in co-rotating twin screw extruders in a reasonable computation time;

ii. To develop/implement multi-objective versions of the most popular metaheuristics to deal with TSCP;

iii. To make a comparative study of the performance of the several metaheuristics applied to TSCPs;

iv. To assess the efficiency of the optimization methodology by studying a practical case studies.

Chapter 3

Modeling of Co-rotating Twin-Screw Extruder *

A global plasticating model for co-rotating twin screw extruders is presented, with the capability of predicting from hopper to die the evolution of pressure, temperature, shear rate, degree of fill, residence time and mechanical power consumption. The modeling software developed only requires moderate computational resources, making it suitable for use with optimization algorithms. The predictions obtained are compared with experimental measurements along the extruder axis for a number of runs involving changes in operating conditions and screw geometry. Globally, the results validate the software and also put in evidence the process steps where improvements in modeling are required.

---

* The contents of this chapter was adapted from: Teixeira, C., Covas, J.A. and Gaspar-Cunha, A., 2012. Flow and Heat Transfer Along the Length of a Co-rotating Twin Screw Extruder. *Polymer-Plastics Technology and Engineering,* 51, 1567-1577.

3.1. Introduction

Intermeshing co-rotating twin-screw extruders are extensively used for polymer compounding, powder coating, food processing and production of pharmaceuticals. They combine easy operation, good output capacity, availability in a wide range of sizes and modular construction. The geometrical flexibility resulting from modularity offers the possibility of adapting the machine characteristics to the pre-requisites of a given manufacture, not only in terms of the series of functions to be carried out (for example, solids conveying, melting, secondary feeding of an additive, mixing, devolatilization, pumping and shaping), but also of their nature (for example, mixing can be made to range from purely distributive to mostly dispersive) [WHI10].

Flow and heat transfer in these machines is complex and rather different from that in single screw extruders, due to the presence of intermeshing zones between the screws, independence between screw speed and output, flow developing in partially filled or in fully filled channels and significant local viscous dissipation. Therefore, in order to better understand the physical, rheological and thermal phenomena developing along these machines, screw pull-out experiments were performed [POT96a, POT96b, BAW98, VER01], following the strategy previously adopted for single screw extruders. An interesting variant to these experiments consisted in making flow visualization studies using glass windows on the barrel or using transparent barrels [WON97, LIU01, CAR02, YIC03]. Based on the knowledge obtained from this research, process modeling attempts have either focused on specific stages, or on the entire sequence of events. The geometrical description of the screw channel and kneading blocks made by Booy [BOO78] paved the way for subsequent flow analyses. The semi-circular channel cross-section is often approximated to a rectangular channel [BOO78, WHI10], while in the kneading blocks the flow is either considered to develop around the individual disks [VER98] or along a channel formed by the kneading block, at an angle in relation to the screw axis defined by the degree of staggering [DEL93]. Solids conveying was modeled taking into consideration partially and fully-filled regions [CAR93, BAW95, POT96b, WON00]. Given its relevance to the process, melt flow along kneading blocks has been the subject of a higher number of studies. Current 3-D simulations use advanced numerical techniques and may be coupled to analyses of the extent of mixing [VAL09, FAR10], but they are usually computationally costly. It has been difficult to establish a conclusive melting mechanism in this type of machines. Bawiskar and White [BAW98] used a model analogous to the Tadmor-type melting model [TAD06], but assuming the simultaneous co-existence of a melt layer near to the barrel surface and solid bed layer in

contact with the screw. Potente and Melisch [POT96a] proposed a model exclusively based on the melting of particles uniformly suspended in the polymer melt. Zhu et al. [ZHU01] and Vergnes et al. [VER01] assumed that this melting mode becomes dominant when at least 50% of the material is molten. Parallely, Gogos and co-workers [GOG98, QIA00, QIA03] demonstrated experimentally that plastic energy deformation could also provide an important contribution for melting but, apparently, the consequent melting model has not yet been developed.

Despite these difficulties, several global modeling programs have been developed, some of them being commercially available. Most adopt relatively simple 1-D flow analyses to describe the process from hopper to die at reasonable computational costs. Vergnes et al. [VER98] put together a program (Ludovic®) that computes melt flow along the extruder. Melt pressure, temperature and residence time predictions have been validated experimentally [CAR00]. Later a particle dispersed melting model was included [VER01]. The software developed by White et al. [WHI01b] (Akro-Co-Twin-Screw) and Canedo [CAN99] (TXS™) consider solids conveying, melting and melt conveying. Similarly, the programme (SIGMA) developed by Potente *et al.* [POT99] covers flow from solids conveying to die exit; depending on the non-linearity of the differential equations to be solved, analytical 1-D or 1.5-D approaches are applied. Wilczynski *et al.* [WIL12] proposed a generalized approach for modeling of screw plastics processing techniques.

This work aims at taking advantage of the previous modeling efforts to develop a more complete and precise calculation of flow and heat transfer along co-rotating twin screw extruders, from hopper to die exit: i) taking on board all the relevant process stages; ii) providing sufficiently accurate predictions for all major process parameters; iii) being sensitive to changes in operating conditions and screw geometry (particularly kneading blocks, where some of the available programs fail); and iv) requiring low computation times (so that the program can be coupled to an optimization algorithm [GAS02, GAS05, POT06]).

This chapter is structured as follows. The mathematical models for the various process steps and the global computer implementation are described in detail in Section 3.2. Considering a specific extruder and an experimental plan that evidences the effect of operating conditions and screw geometry on its behavior (presented in Section 3.3), the predictions are confronted with experimental measurements in Section 3.4. Finally, the major conclusions are presented in Section 3.5.

3.2. Process Modeling

3.2.1. Geometry and flow kinematics

In twin screw extrusion, screws are usually assembled by putting together a certain number of individual elements that are selected from a larger pool exhibiting variations in pitch, helix or staggering angle and length or number of kneading disks. Often, the barrel can also be constructed by coupling individual segments. The following possibilities will be studied here:

i) *Right handed elements*, with a positive helix angle and forward conveying capacity; the lower the angle, the more efficient the transport (see Figure 3.1);

ii) *Left handed elements*, with a negative helix angle, imposing a flow restriction that is useful for mixing (both distributive and dispersive) and/or melting (see Figure 3.2);

iii) *Kneading blocks* comprise a variable number of kneading disks that can be staggered at positive, neutral (i.e., 90°), or negative angles (see Figure 3.3). In the first case, they convey the material forward while inducing some degree of distributive mixing, while negative angles impose a flow restriction that, similarly to left handed elements, can cause melting and/or considerable dispersive mixing. Neutral blocks have no conveying characteristics and are quite efficient for mixing.

Figure 3.1a illustrates the flow along a right handed double flighted screw element. Assuming the customary simplification of having a stationary screw and a rotating barrel with velocity V_b, the down-channel and cross-channel components are identified in Figure 3.1b. Under normal starve-feed conditions these elements work partially filled. Conversely, in the case of left handed elements (Figure 3.2) the restriction imposed to the flow generates a back pressure. For both geometries, it makes sense to replace the circular-like cross-channel geometry by a rectangular channel with the same area, as done in previous studies [BOO78, POT96b, VER98, WHI10]. As seen in Figure 3.1c and 3.2c, the height, H, was fixed as $H_{\max}/2$ and the width, W, was calculated. Figure 3.3 deals with the geometry of a kneading block (with a constant staggering angle and all disks with identical thickness). Figure 3.3a represents a front view and defines the staggering angle, α. Figure 3.3b presents a side view and identifies the two flow channels that are formed by the staggering, with flow rates Q_C and Q_L, while in Figure 3.3c a sequence of three kneading disks are represented

unwrapped. The width of each channel changes with α. As before, an approximation to a rectangular channel is appropriate [BOO78, DEL93].

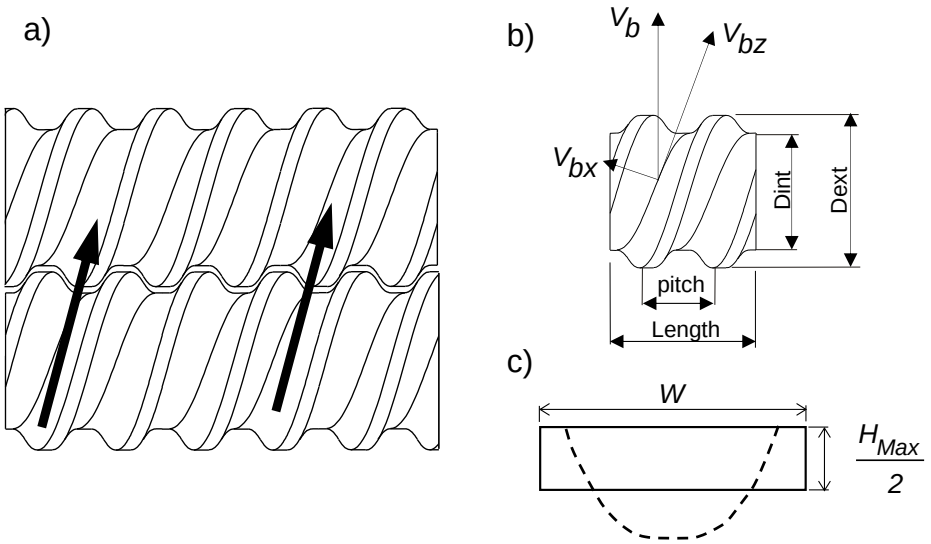

Figure 3.1. Geometry and flow kinematics of a right-handed screw element. a) material flow; b) barrel velocity components; c) simplification of channel cross-section.

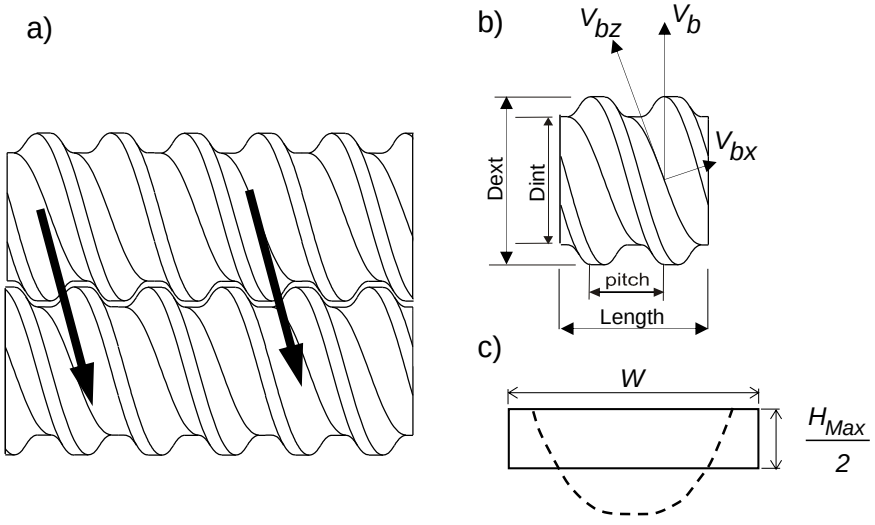

Figure 3.2. Geometry and flow kinematics of a left-handed screw element. a) material flow; b) barrel velocity components; c) simplification of channel cross-section.

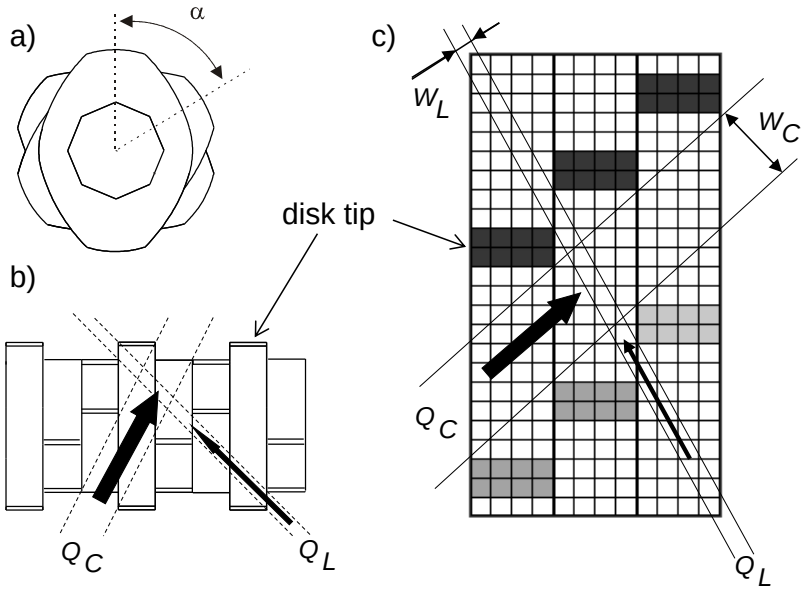

Figure 3.3. Geometry and flow kinematics of a kneading block: a) front view; b) side view; c) a series of three unwrapped disks.

3.2.2. Individual Process Steps

For a given operation, the nature and extent of every individual step depends on the screw profile and operating conditions. Experiments showed that as soon as the material approaches the first restrictive zone pressure raises, heat transfer becomes more efficient and melting is initiated [POT96a, VER98, BOO80, WHI10]. Also, the combined effect of heat conducted and generated by friction and deformation induces quick melting [POT96a, VER98, WHI10]. This means that even for a simple screw profile, consisting of conveying elements and one restrictive zone, the following succession of plasticating steps develops [MAC99, YIC03]:

1) solids conveying without pressure in the initial screw turns;
2) solids conveying under pressure as the material comes sufficiently close to the restrictive element;
3) melting;
4) melt conveying under pressure along the restrictive element;
5) melt conveying without pressure;
6) melt conveying under pressure in the final screw turns, due to the presence of the die.

Figure 3.4 identifies the typical location and extent of these steps, together with the corresponding axial pressure profile, for a more practical screw profile containing two restrictive/mixing zones. The

pressure drop along a restrictive element must be balanced by pressure generation upstream. As a result, at least part of conveying elements upstream of restrictive zones may work fully filled. Consequently, melting may be completed upstream of the first restrictive block.

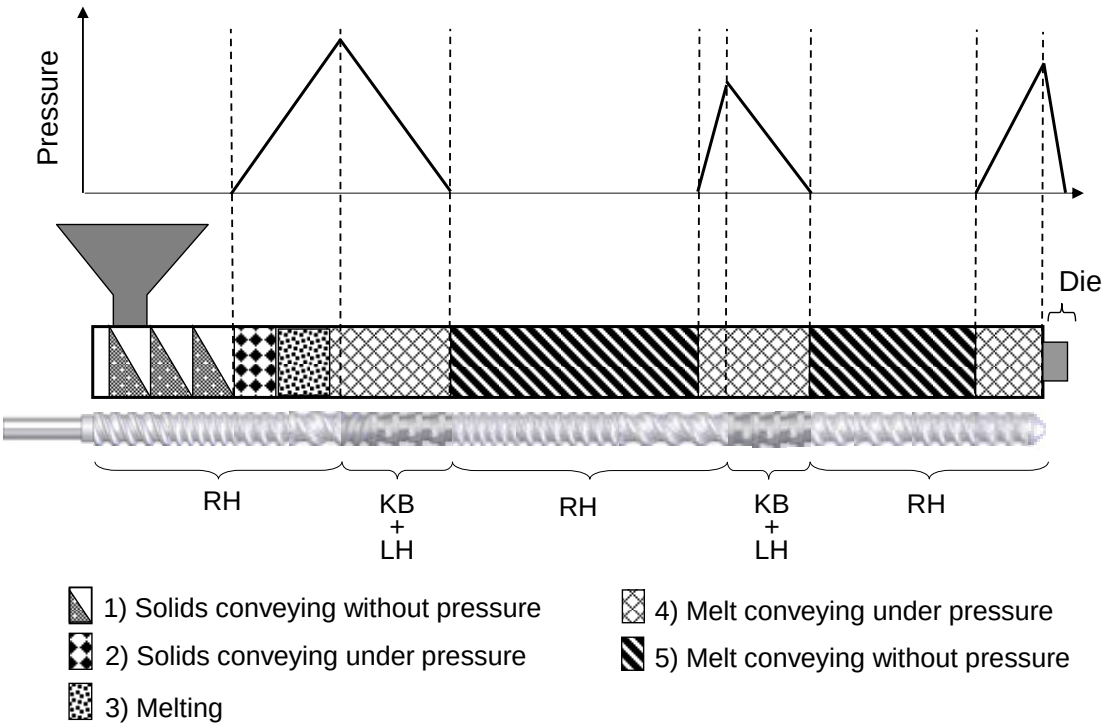

Figure 3.4. Individual process steps and corresponding axial pressure profile for a screw containing two restrictive/mixing zones (RH: right handed elements; LH: left handed elements; KB: kneading block).

Figure 3.5 represents the physical models corresponding to steps 1) to 5) identified above. Material is fed to the screws at room temperature, at a rate set by a feeder. In turn, the screw speed will determine the average degree of fill. The incoming material is conveyed forward in partially filled channels by right handed elements and should remain relatively cold, due to poor transfer/generation conditions (step I). Here, residence time and fill ratio can be computed, as done by Vergnes et al. [VER98]. Channel filling is induced in due course by the presence of restrictive elements (their restrictive character is determined by the geometry and operating conditions), which reduce the conveying capability of the element under analysis (step II). As pressure develops, thermal exchanges and dissipation are favored. The analysis of Tadmor and Broyer for single screw extruders should remain valid here [TAD72]. Pressure development is computed from force and momentum balances on a differential down-channel increment. The temperature profile is estimated by solving the energy equation, taking into account heat conducted from the barrel and heat dissipated at barrel and screw root due to friction. The former is calculated

in the solids and barrel directions, while in the second the screw is assumed as adiabatic. The mechanical power consumption includes the power dissipated at the barrel, screw root and flights and the energy needed to compact the polymer. The average residence time is the ratio between the down-channel length and the average solid bed velocity. Although, as discussed above, a conclusive melting mechanism has yet to be established, it seems logical to admit that increasing solids compaction, heating from the barrel and higher friction dissipation also at the barrel should promote quick local melting and formation of a melt film (step III). For example, Vergnes et al. assumed the co-existence of a solid plug and a melt film near the barrel wall [VER01]. Flow in the melt film is taken as fully developed in the down and cross channel directions. In turn, the energy equation includes heat convection, conduction and viscous dissipation [KAC72], while computation of the pressure profile takes into account that viscous forces exist at the barrel surface. The linkage between melt film and solid bed is made via mass and heat balances at the solid bed-melt film interface.

As pointed out earlier, previous experimental studies of melting in co-rotating twin screw extruders generated ambiguous conclusions, with some authors observing the progressive melting of a solid plug and melt accumulating in a melt pool [BAW98, LIU01, YIC03], while others reported the progressive conversion of a solid-rich suspension into a melt-rich suspension [POT96a, VER01, ZHU01]. At this point, it is worth reminding the early stages of melting in a single screw extruder, where a delay zone also forms (i.e., melt film(s) separate the solid plug from the channel wall(s)), before the well-known Tadmor melting mechanism develops.

Predictions generally overestimate the extent of this zone, which has been interpreted as evidence of the infiltration of the freshly formed melt in between the polymer pellets [AGA91]. The intensity of this phenomenon is probably related with the local pressure, pellet shape, size and stiffness. These parameters could also trigger different melting sequences in the case of the twin screw extruder. In order to take into account the various observations reported and guarantee physical coherence with the preceding delay zone (step III) - extending until the polymer near the screw surface melts - an hybrid melting mechanism was adopted

This makes possible the use of mathematical models developed for single screw extrusion, in this case the 5-zone melting model proposed by Elbirli et al., which also presumes the existence of melt films near the barrel, screw root and screw flight (step IV) [ELB84]. The simple shape of the solid bed depicted in the figure is certainly an approximation, as it should become increasingly distorted.

This, together with the raising pressure and shear stress resulting from the flow of a progressively more abundant melt, will eventually break the solid bed (back into the original pellets) and create a suspension. Thus, when the amount of melt exceeds 50% of the material, a dispersive melting model is activated (step V), assuming a uniform distribution of spherical pellets in the melt (not far from closed packing) and that their temperature increases by heat conducted from the melt [VER01, ZHU01]. Melt flow is computed by solving the momentum and energy equations, allowing for viscous dissipation and heat conduction, with the viscosity being described by the Carreau-Yasuda law for concentrated suspensions [VER01]. Melt flow in fully-filled sections (step VI) and in the die is calculated as 2-D non-isothermal. In the case of kneading blocks, flow is initially computed exclusively in the larger channel (see Figure 3.3), in order to estimate the shear rate (and compute viscosity). Q_C and Q_L are determined from a mass balance and solving the corresponding governing equations. The average shear rate along the block is taken as a weighted average in the two channels. Finally, pure drag flow develops in partially filled channels (step VII) and pressure is nil. Melt temperature is computed via a thermal balance, considering heat conduction at the barrel surface and heat dissipation at all surfaces [VER98].

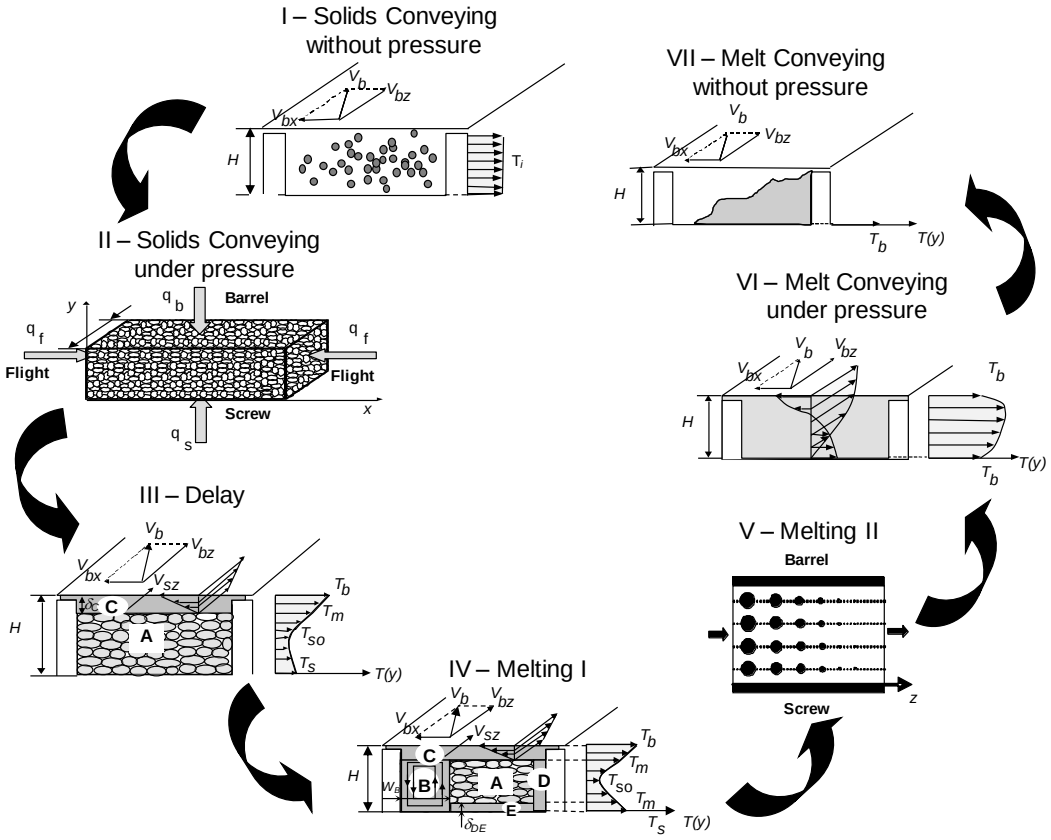

Figure 3.5. Models for each plasticating step.

3.2.3. Global algorithm

The individual steps presented above were linked together into a global model by means of coherent boundary conditions - see algorithm flowchart in Figure 3.6.

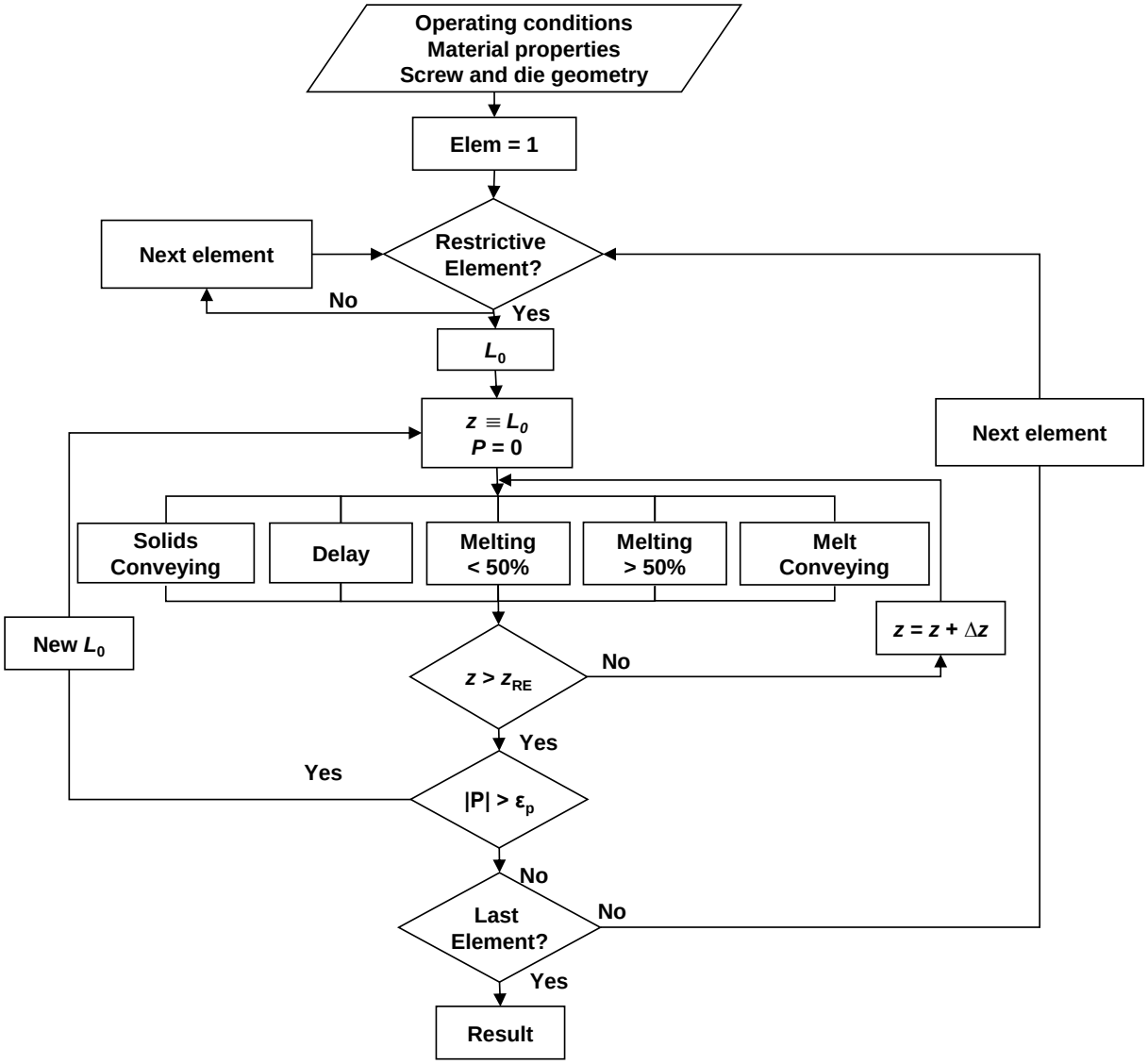

Figure 3.6. Flowchart of the global modeling program developed.

Input data include operating conditions (screw speed, output, barrel/die temperature profile), material properties, screw and die geometries. The calculations proceed from the screw inlet towards the die.

The program starts by identifying the restrictive screw elements. Then, it sets off an iterative procedure to establish the screw location (L_o) upstream of the first restrictive element where the channel becomes fully filled (as illustrated in Figure 3.7). An initial L_o, where the pressure is set as

nil, is randomly attributed. For small Δz screw down-channel increments, the program calculates the evolution of the main process parameters up to the element periphery. If the pressure here is not nil (in practice, if it is higher than a numerical difference, ε_p), L_o is incremented or decremented and the process is repeated until the pressure becomes sufficiently small. Melting should take place in this first restrictive element. Therefore, when considering the various Δz down-channel increments, the sequence of solids conveying, delay, melting and melt conveying is tested out. An identical iterative procedure is followed for all subsequent restrictive elements as well as the die, but now always assuming melt conveying. In between restrictive zones, the evolution of melt temperature is also computed.

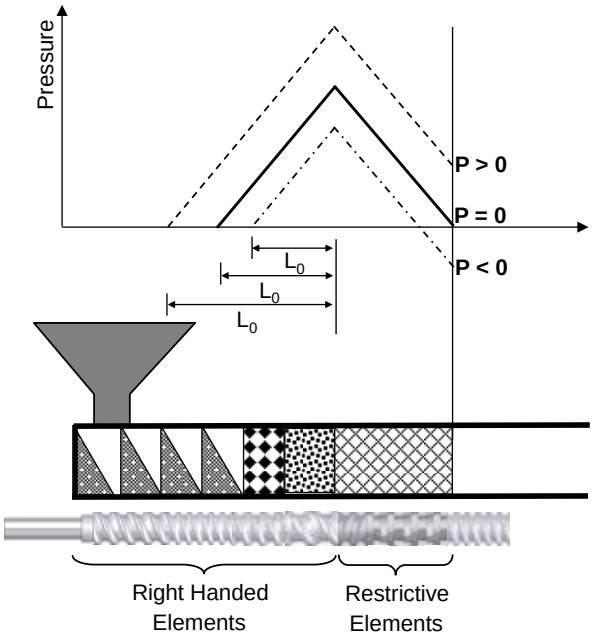

Figure 3.7. Iterative procedure to find the location upstream of a restrictive zone where the channel becomes fully-filled.

3.3. Equipment, material and experimental procedure

The predictions produced by the modeling package were directly confronted with experimental measurements. A laboratory Leistritz LSM 30.34 intermeshing co-rotating twin screw extruder (screw diameter of 30 mm, centerline distance 34 mm, L/D = 29) was fitted with a number of sampling devices along its length (Figure 3.8), at positions where the screws are expected to work fully filled. Each device takes around one second to remove a sample from within the extruder

[MAC99], thus allowing a correct observation of its physical appearance (solid, partially molten or fully molten). Moreover, sticking a fast response thermocouple into the bulk of the sample prior to its removal provides a good measurement of the average temperature [CAR04]. Figure 3.8 also illustrates the screw profile, comprising 16 elements that create three different restrictive zones, with the geometrical characteristics described in Table 3.1.

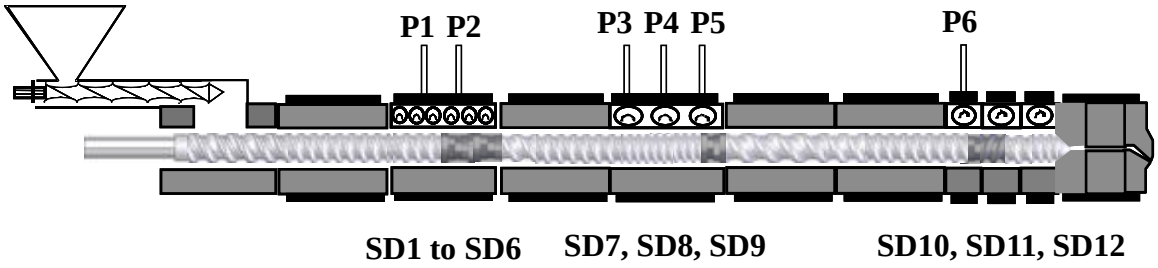

Figure 3.8. Layout of the Leistritz LSM 30.34 co-rotating twin screw extruder, with 12 sampling devices (SD1 to SD12) and 6 melt pressure transducers (P1 to P6).

Table 3.1. Screw profile used as reference (a negative pitch indicates a left handed element and KD-30° denotes a block of kneading discs staggered -30°).

Screw element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Length (mm)	97.5	120	60	30	60	30	120	60	30	120	60	60	22.5	30	30	30
Pitch (mm)	45	30	30	30	KD -30°	60	30	20	KD -60°	60	45	30	KD 30°	-20	30	20

Using a pigment as a tracer fed directly to the screws at time zero, the average total residence time was taken as the time required to detect visually changes in the color of extrudate cross-sections collected every ten seconds. This is of course a crude estimation, but since previous experiments have demonstrated that in co-rotating twin-screw extruders the evolution of tracer concentration with time exhibits an initial sharp increase up to a maximum followed by a long tail, the average residence time should not be too different from the minimum residence time [CAR04].

A polypropylene homopolymer (ISPLEN PP030 G1E, manufactured by REPSOL) was used in the experiments. It is an extrusion grade - MFI of 1.75 (230°C/2.16 kg) - with the properties assembled in Table 3.2. The viscosity is described by the Carreau-Yasuda law using the data obtained in a Rosand RH8 Dual Capillary Rheometer, considering both the Bagley and Rabinowistch corrections:

$$\eta = \eta_0 a_T \left(1 + \left(\hat{\lambda} a_T \dot{\gamma}\right)^a\right)^{\frac{n-1}{a}} \tag{3.1}$$

with

$$a_T = \exp\left(\frac{E}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right) \tag{3.2}$$

Table 3.2. Properties of the polypropylene homopolymer (ISPLEN PP030 G1E from REPSOL).

			PP (ISPLEN PP 030 G1E)	
Density	Solids	ρ_s	690.90	$kg.m^{-3}$
	Melt	ρ_m	902.00	
Thermal Conductivity	Solids	k_s	0.21	$W.m^{-1}{}^{o}C^{-1}$
	Melt	k_m	0.18	
Specific Heat	Solids	C_s	1881.92	$J.kg^{-1}$
	Melt	C_m	1974.55	
Melting	Heat	H_m	89.49x10³	$J.kg^{-1}$
	Temperature	T_m	170	^{o}C
Viscosity Carreau-Yasuda law		η_0	3041.48	$Pa.s$
		E/R	4023.29	K
		$\hat{\lambda}$	0.17	s
		a	1.82	
		n	0.35	
		T_0	493.15	K

A Perkin Elmer DSC 7 was used to determine the specific heat, the melting heat and the melting temperature. The remaining properties were taken from the literature.

The experimental/computational plan presented in Table 3.3 attempts to evaluate the sensitivity of the extruder and software to changes in operating conditions (screw speed, output and set temperature, T_b) and screw configuration. Six screw configurations were tested, (A to F in Table 3.3.), as seen in Table 3.4. Screw A is considered the reference screw (geometrical characterization

in Table 3.1). In screws B to D the restrictive zone upstream have distinct geometries, in order to investigate their influence on melting. In turn, the different geometries of the second mixing zone of Screws E and F provide the possibility to ascertain their effect on melt flow.

Table 3.3. Experimental / Computational plan.

Run	Screw Configuration	N (rpm)	Q (kg/h)	T_b (ºC)
1	**A**	**150**	**8**	**220**
2	A	**100**	8	220
3	A	**200**	8	220
4	A	150	**4**	220
5	A	150	**12**	220
6	A	150	8	**205**
7	A	150	8	**235**
8	**B**	150	8	220
9	**C**	150	8	220
10	**D**	150	8	220
11	**E**	150	8	220
12	**F**	150	8	220

Table 3.4. Screw configurations (LH denotes a left handed element).

Screw Configuration	Screw Element Changed (Table 3.1)	Type of Screw Element
B	5	KD -60°
C	5	KD 90°
D	5	KD 90° + LH (Pitch=30mm)
E	9	KD -30°
F	9	LH (Pitch=30mm)

3.4. Results and discussion

3.4.1. Global Performance

Figure 3.9 shows the experimental and the predicted pressure and temperature profiles along the extruder for a single run (run 1 – Table 3.3). As expected, and due to the imposed constraints, the pressure starts to develop in the screw element upstream of the various restrictive modules (numbers 5, 9 and 13-14 in Table 3.1) and before the die, while the maximum is reached at the start of the restrictive elements and die. The temperature increases in filled channels due to the more efficient heat conduction from the barrel and to viscous dissipation. This situation ceases during flow in partially filled channels, where the average temperature decreases and approaches the set value.

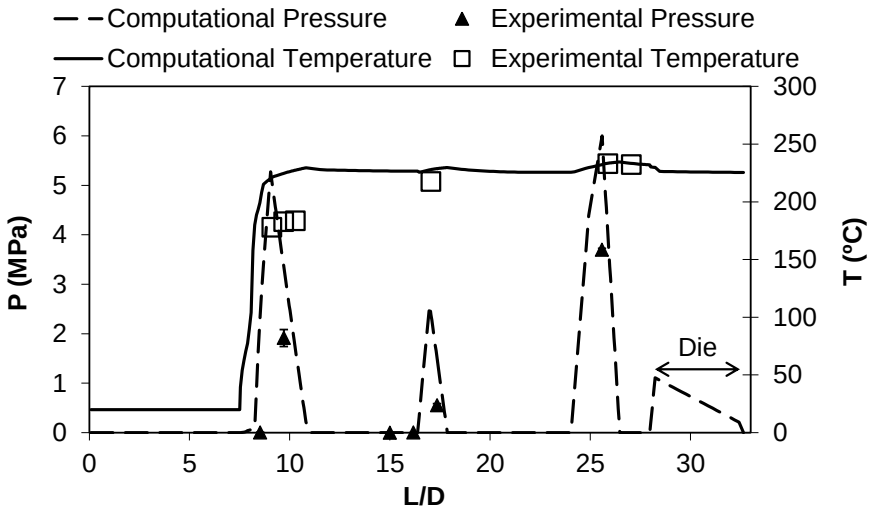

Figure 3.9. Pressure and temperature evolution along the screws for run 1.

The experimental measurements are generally in good agreement with the predictions, with the exception of the three temperature readings upstream. As observed earlier during the validation of other computational predictions for twin screw extrusion [CAR00], the pressures seem somewhat underestimated, but this could be influenced by the very sharp axial gradients, which originate large differences even for small axial increments. Also, it is worth noting that despite the specific location of the devices in the extruder, the measurements and material collection are influenced by the flow over a certain axial distance (depending on the local geometry) both upstream and downstream of that location, due to the screw rotation. Thus, a sort of horizontal span (up to ±0.6L/D) should be considered when observing the data. Even so, the above referred temperatures are undoubtedly

lower than the predictions. They were measured in the material collected from the devices SD04 to SD06 (Figure 3.8), as the channels upstream were only partially filled (this confirms that the melt pressure here is nil). At SD04 solids and melt were present, at SD05 the percentage of melt seemed to have increased, while at SD06 the sample was essentially molten. Melting is predicted to develop very fast, starting at L/D = 7.99 and being completed by L/D = 8.78. These values should be compared with L/D = 8.6 (location of SD03, where the channel was partially filled and most probably the material was solid) and L/D = 10.3 (location of SD06, where the channel if full and the material is totally melted), respectively. Therefore, the melting model implemented seems to overestimate the melting rate, and this is why the local measured temperatures are lower than the predicted ones.

Figure 3.10 presents the local cumulative residence time, for the same run of Figure 3.9 (run 1). The prediction compares satisfactorily with the experiment (single point in the figure, with a resolution of about 10 seconds). As expected, local residence times are higher at restrictive elements, while at conveying zones the higher the pitch the lower the slope of the axial evolution (for example, between L/D = 11 and 14 the rate of the cumulative residence time increase is higher than between L/D = 18 and 23, where the pitches are 30/20 and 45/30, respectively). Similarly, the figure shows that the geometry of the mixing zones (in this case, KD-30, KD-60, KD30+LH-20) also influences the local residence times. This will be discussed below in greater detail. Figure 3.10 also portrays the axial evolution of the fill ratio (volume occupied by the material relative to the total volume).The experimental points were obtained using the sample collecting devices, as it is quite easy to identify whether the channels operate fully filled (the collecting chamber fills quickly), nearly filled (the collecting chamber fills partially for the same collecting time), or moderately filled (very little material is collected under the same conditions). In the graph, these situations correspond to fill ratios of 1, 0.5 and 0.1, respectively. The matching between predictions and observations is quite good, except for the beginning of the restrictive zone upstream, due to the difficulties discussed above with modeling of melting.

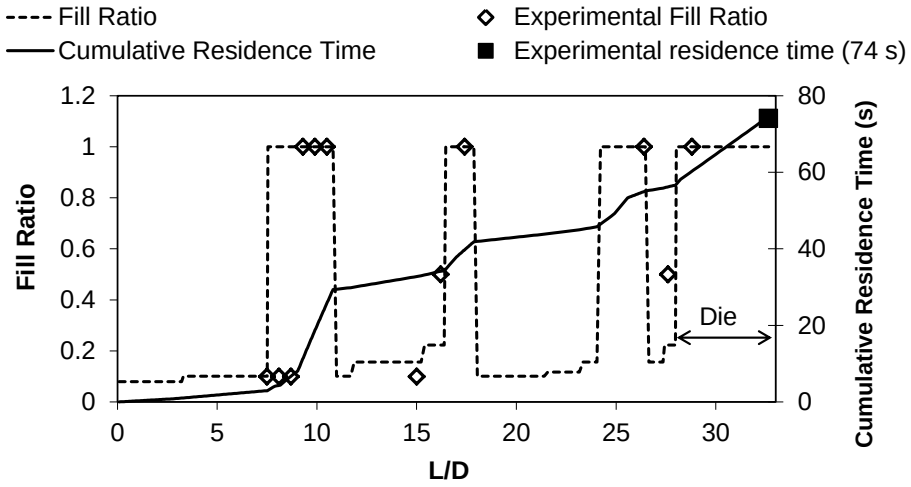

Figure 3.10. Local and cumulative residence times and fill ratio (run 1).

3.4.2. Effect of operating conditions

Figure 3.11 and Table 3.5 show the effect of screw speed on the response of the extruder (runs 1 to 3 in Table 3.1). The maximum pressure always occurs at the beginning of each restrictive zone, but the value decreases with increasing screw speed as the extruder pushes forward the same amount of material but at higher rates, reducing the viscosity and lowering the flow restriction. A higher pressure generation causes an increase of the number of fully filled channels upstream of any restrictive zone. As seen in Table 3.5, this raises the global residence time for smaller screw speeds, despite the constant output. The differences between predictions and computations are generally below 10%.

As expected, the higher the screw speed, the higher the average shear rate (Table 3.5). In fully filled channels, the rates are circa two times higher than those in partially filled regions. The maximum shear rate is reached in the gap between the barrel and the tip of kneading discs. Its value is calculated to be roughly 25% higher than the average in the channel and also increases with screw speed. In addition, despite the rheo-fluidifying character of the polymer melt, higher screw speeds are associated to higher viscous dissipation. The computations follow the experimental trend, even if the predicted temperature variation is smaller than the measured one (in Table 3.5, the increase in temperature at 150 and 200 rpm is given relative to the value at 100 and 150 rpm, respectively). A discrepancy between both was not unexpected, as temperature measurements were made on samples that included melt from the channels and from the gap

between the barrel and the tip of the kneading discs, which should be hotter due to the local higher shear rates.

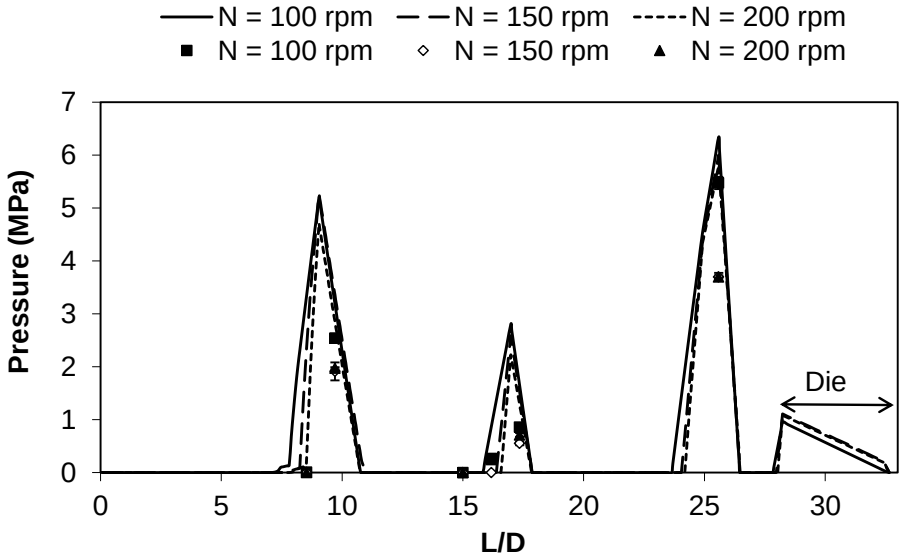

Figure 3.11. Effect of screw speed on pressure and temperature evolution along the screws.

Finally, Table 3.5 also demonstrates that screw speed influences the evolution of melting. Although the length of the fully filled region upstream of the first restrictive zone diminishes with increasing screw speed, the location of the melting onset shifts progressively downstream due to the lower residence time for heat transfer. However, once started, melting tends to be somewhat faster for higher screw speeds due to the growing contribution of friction (melting I stage) and viscous dissipation (melting II stage). The experimental validation of these predictions is difficult, as samples can only be collected at fixed axial distances along the barrel and the changes caused by varying the screw speed seem to be generally smaller than the length between adjacent sampling ports. Still, the trend seems correct.

Figures 3.12 and 3.13 and Table 3.6 illustrate the effect of changing the mass output on the same process variables (runs 1, 4 and 5 in Table 3.1). In general, the experimental data corroborate the predictions, even though the effect on fill ratio is difficult to identify experimentally. Anyway, the sampling device SD03 (at L/D = 8.6) was nearly empty when operating at 4 and 8 kg/h and partially filled for 12 kg/h, while SD11 (at L/D = 27.1) was partially filled for 4 kg/h and 8 kg/h and totally filled for 12 kg/h, which matches well the calculations. Not only the degree of filling in partially filled channels is predicted to increase, but the length of the fully filled regions upstream of restrictive zones should become longer (Figure 3.12 and Table 3.6). Therefore, as the output

increases efficient heat transfer from the barrel should start a little earlier in the first restrictive zone, but the quantity of material to be melted augments. As seen in Table 3.6, the combined effect of these phenomena results in an advance of the melting onset and a decrease of the melting rate with increasing output. As expected, the global residence time decreases with increasing output. Both the experimental data and the computations in Table 3.6 point to a decrease of the average melt temperature with increasing output, which is evidently linked to the decrease in residence time and is in good agreement with previous similar measurements [CAR00]. The decrease of the maximum shear rate and of the average shear rate in the fully filled channels with increasing output is explained by the changes in the shape of the velocity profile. The latter comprehends a positive drag component and a negative pressure component. As the output rises, the whole profile flattens. As expected and seen in Figure 3.13, melt pressures increase with output and the channels upstream fill progressively earlier. The raise in pressure with output is relatively moderate, due to the strong non-Newtonian character of the polymer melt.

Table 3.5. Influence of screw speed on process parameters (Comp.: Computational; Exp.: Experimental)

			Screw speed (rpm)		
			100	**150**	**200**
Cumulative Residence time (s)		Comp.	95.6	74.5	60.1
		Exp.	80.0±5	74.0±5	69.0±5
Shear rate (s^{-1})	Fully filled	Comp.	83.0	125.4	158.0
	Partially-filled	Comp.	42.5	63.9	85.3
	Maximum	Comp	100.9	179.3	222.7
Increase in average melt temperature (ºC)		Comp.	–	1.8	1.2
		Exp.	–	8.0	6.7
Melting (L/D):		Comp.	0.79	0.79	0.70
i) length needed		Exp.	0.9±0.3	0.9±0.3	0.6±0.3
ii) location of the melting onset		Comp.	7.54	7.99	8.25
		Exp.	9.3±0.6	9.3±0.6	9.3±0.6
iii) filled length upstream of the first restrictive element		Comp.	2.06	1.53	1.36

Table 3.6. Influence of output (Comp.: Computational; Exp.: Experimental).

			Output (kg/hr)		
			4	**8**	**12**
Cumulative Residence time (s)		Comp.	98.2	74.5	66.5
		Exp.	95.0±5	74.0±5	55.0±5
Shear rate (s^{-1})	Fully filled	Comp.	150.7	125.4	110.2
	Partially-filled	Comp.	64.1	63.9	63.5
	Maximum	Comp.	221.1	179.3	132.8
Increase in average melt temperature (ºC)		Comp.	–	-1.0	-0.9
		Exp.	–	-11.7	-4.2
Melting (L/D):		Comp.	0.44	0.79	1.06
i) length needed		Exp.	0.6±0.3	0.9±0.3	1.2±0.3
ii) location of the melting onset		Comp.	8.07	7.99	7.72
		Exp.	9.3±0.6	9.3±0.6	9.3±0.6
iii) filled length upstream of the first restrictive element		Comp.	1.18	1.53	1.98

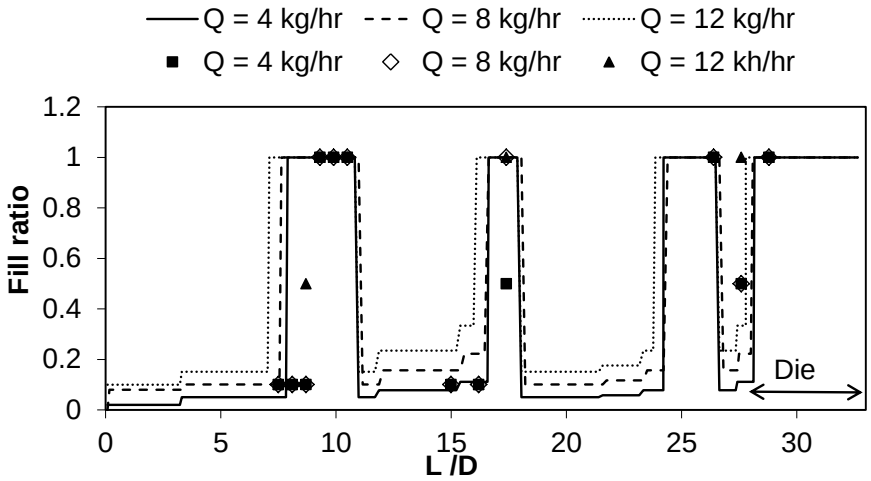

Figure 3.12. Effect of output on fill ratio.

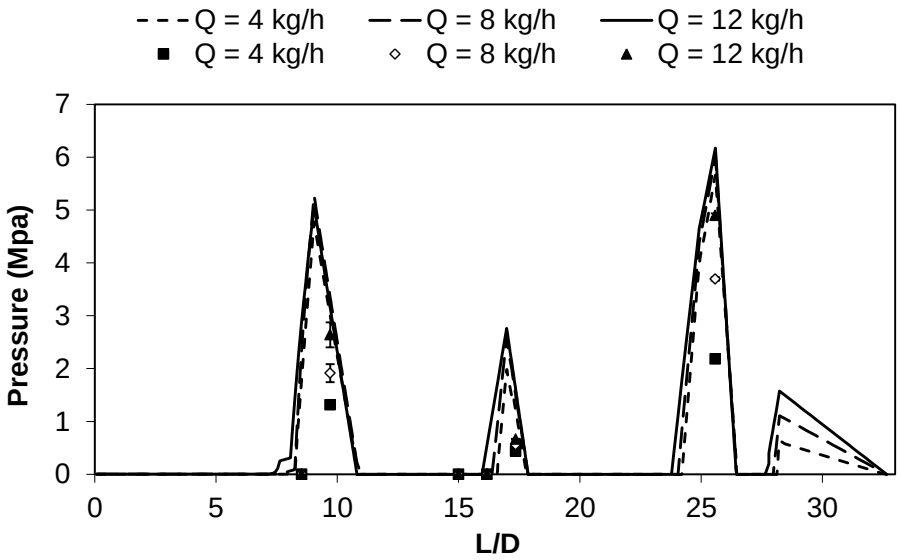

Figure 3.13. Effect of output on pressure and temperature evolution along the screws.

The effect of barrel temperature is presented in Table 3.7 (runs 1, 6 and 7 in Table 3.1). As the set temperature increases so does the melt temperature, which yields a moderate effect on residence time - the resulting lower viscosity levels facilitate progress along the extruder. Similarly, melting is promoted, not only in terms of the location of the onset, but also on its extension.

Table 3.7. Influence of barrel temperature profile (Comp.: Computational; Exp.: Experimental).

			Barrel Temperature (ºC)		
			205	**220**	**235**
Cumulative Residence time (s)		Comp.	76.8	74.5	71.8
		Exp.	75.0±5	74.0±5	67.0±5
Shear rate (s^{-1})	Fully filled	Comp.	126.2	125.4	125.7
	Partially-filled	Comp.	63.9	63.9	63.9
	Maximum	Comp.	178.6	179.3	179.6
Increase in average melt temperature (ºC)		Comp.	–	14.0	13.8
		Exp.	–	6.7	17.8
Melting (L/D):		Comp.	1.05	0.79	0.61
i) length needed		Exp.	1.5±0.3	0.9±0.3	0.6±0.3
ii) location of the melting onset		Comp.	8.07	7.99	7.89
		Exp.	9.3±0.6	9.3±0.6	9.3±0.6
iii) filled length upstream of the first restrictive element		Comp.	1.79	1.53	1.36

3.4.3. Effect of Screw Geometry

In this section, the effect of changes in the geometry of the restrictive elements will be investigated both in terms of melting and melt conveying characteristics. For this purpose, the restrictive elements of screw configuration A (Table 3.1) located at positions 5 (melting region) and 9 (melt conveying region) were changed as shown in Table 3.3.

Figures 3.14 and 3.15 and Table 3.8 refer to the behavior in the melting region. As demonstrated in Figure 3.14, the geometry influences significantly the onset of temperature development, due to the effect on the axial extent of the fully filled region upstream the first restrictive element. The calculated ranking of quicker temperature development is: KD90+LH > KD-60 > KD-30 > KD90. Coherently, this is also the ranking of the peak pressures shown in Figure 3.15. The real pressure readings follow roughly the same order, with: KD90+LH > KD-30 ≈ KD-60 > KD90 (the pressure differences seen in the pressure peaks downstream, where no changes in processing conditions were generated, indicate the order of magnitude of the associated error - circa 1MPa).The practical temperature readings were unable to discriminate the effect of the screw geometry, although it is clear that KD90 induces a slower temperature increase. This is obviously explained by the higher solids content of the samples.

The ranking of the predicted cumulative residence time follows approximately the same order, with a maximum for KD90+LH and a minimum for KB90. The small precision of the corresponding measurements hinders the possibility to discriminate between the various screw profiles, but the maximum value seems to occur for KD90+LH. Coherently, the axial length of filled channels upstream the first restrictive element and the location of the melting onset follow similar trends.

As far as melting is concerned, the calculated and measured trends are similar, KB-30 showing the highest efficiency and KB90 the lowest. The results regarding shear rate and relative increase in average melt temperature are more difficult to interpret, especially for KD90 and KD90+LH. Modeling flow along a neutral kneading block and predicting the location where the channels become fully filled is not easy, and has not been clearly tackled in the literature. Following previous experimental observations [POU01], it was assumed that for KB90 only half of the length of the zone would operate fully filled. Obviously, this hypothesis affected the location of the maximum pressure (see Figure 3.15, where the location of the peak pressure for KB90 is different from that of the remaining) and, consequently, the value of the related process parameters, such as the filled length upstream of the first restrictive element, which is nil.

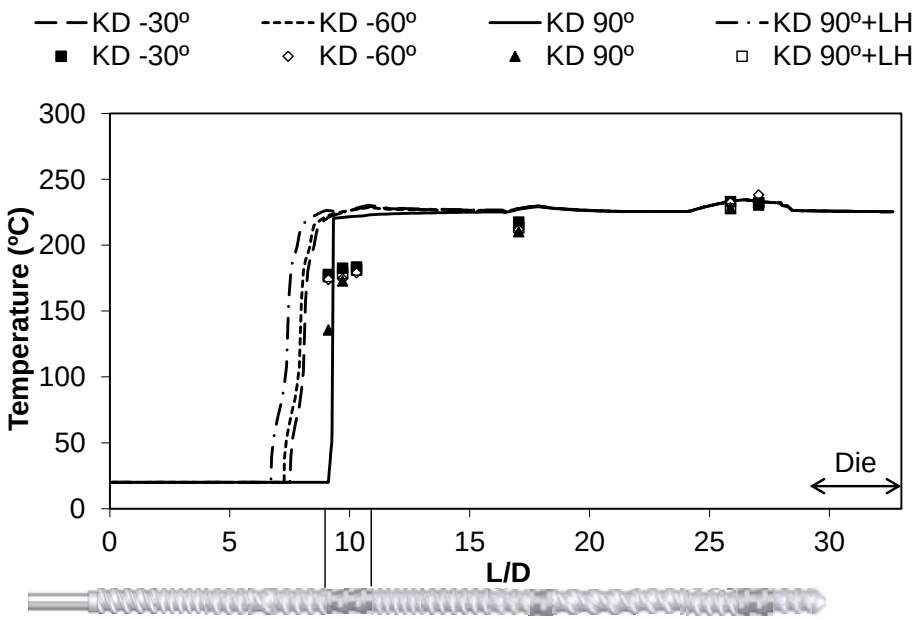

Figure 3.14. Effect of screw geometry (restrictive zone upstream) on temperature evolution along the screws.

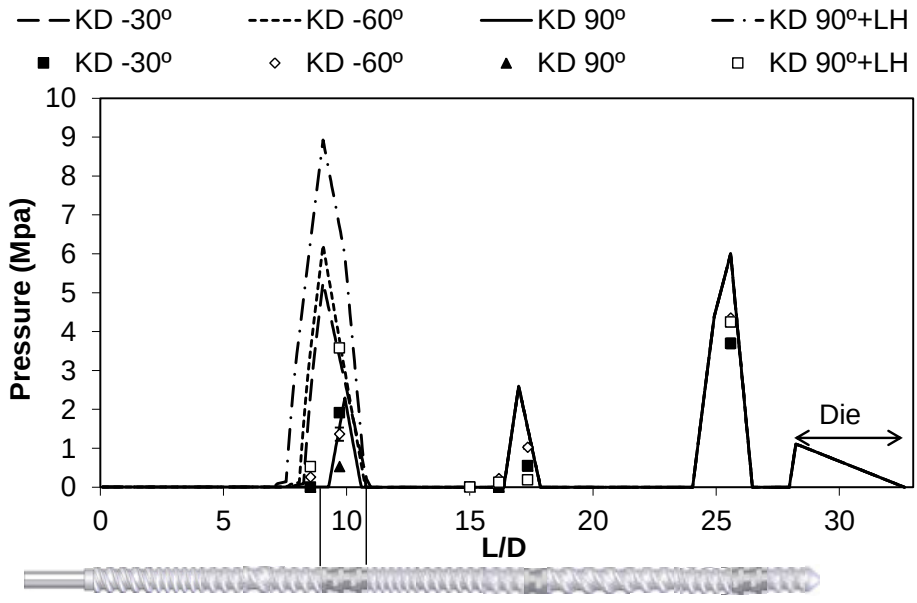

Figure 3.15. Effect of screw geometry (restrictive zone upstream) on pressure evolution along the screws.

Table 3.8. Influence of screw geometry in the restrictive zone upstream (Comp.: Computational; Exp.: Experimental).

			Screw element in the melting (zone 5)			
			KB-30	**KB-60**	**KB90**	**KB90+LH**
Cumulative Residence time (s)		Comp.	74.5	61.4	53.4	76.1
		Exp.	74.0±5	70.0±5	75.0±5	80.0±5
Shear rate (s^{-1})	Fully filled	Comp.	125.4	129.1	201.8	175.9
	Partially-filled	Comp.	63.9	63.9	41.4	63.9
	Maximum	Comp.	179.3	179.1	398.8	400.1
Increase in average melt temperature (ºC)		Comp.	–	0.9	0.2	0.8
		Exp.	–	4.6	2.4	5.4
Melting (L/D):		Comp.	0.79	0.88	1.43	0.88
i) length needed		Exp.	0.9±0.3	1.2±0.3	1.5 ±0.3	1.2±0.3
ii) location of the melting onset		Comp.	7.99	7.80	9.10	7.30
		Exp.	9.3±0.6	9.3±0.6	9.3±0.6	9.3±0.6
iii) filled length upstream of the first restrictive element		Comp.	1.53	1.80	0.00	2.33

Table 3.9 shows the effect of changing the geometry of some screw elements in the conveying region (location SD9, see Figure 3.8). The computations and experiments show the same trend for the relative effect of KB-60, KB-30 and Left Handed elements on the cumulative residence time and increase in average melt temperature.

Table 3.9. Influence of screw geometry on melt flow (Comp.: Computational; Exp.: Experimental).

			Screw element in the melting (zone 9)		
			KB-60	**KB-30**	**LH**
Cumulative Residence time (s)		Comp.	74.5	60.2	76.1
		Exp.	74.0±5	72.0±5	82.0±5
Shear rate (s^{-1})	Fully filled	Comp.	125.4	128.6	133.3
	Partially-filled	Comp.	63.9	63.9	63.6
	Maximum	Comp.	179.3	179.3	179.3
Increase in average melt temperature (ºC)		Comp.	–	0.3	1.1
		Exp.	–	3.9	4.4

3.5. Conclusions

A global plasticating modeling software for co-rotating twin screw extruders, requiring moderate computational resources (1 to 2 minutes in a personable computer – Pentium M 1.75 GHz), has been presented. Several predictions were directly confronted with experimental data obtained along the extruder axis, enabling an investigation of the influence of operating conditions and screw geometry on the machine behavior. The process parameters studied included melt pressure, (solids and melt) temperature, cumulative residence time, fill ratio and average shear rate.

Despite the difficulty in obtaining sufficiently accurate experimental data, in general the latter validated the predicted trends, giving practical utility to the software. The study also pointed to the need to improve the modeling of some process facets, such as melting (with is currently overestimated) and flow in neutral kneading blocks.

Chapter 4

Two-Phase Local Search Algorithm*

The twin-screw configuration problem consists in defining the location of a set of pre-defined screw elements along the screw axis in order to optimize different, typically conflicting objectives. In this work, a simple yet effective stochastic local search algorithm (SLS) for this problem is presented. The algorithm is based on efficient single-objective iterative improvement algorithms, which have been developed by studying different neighborhood structures, neighborhood search strategies, and neighborhood restrictions. These algorithms are embedded into a variation of the two-phase local search (TPLS) framework to tackle various bi-objective versions of this problem. An experimental comparison with a previously proposed multi-objective evolutionary algorithm shows that a main advantage of the SLS algorithm developed is that it converges faster to a high-quality approximation to the Pareto front.

---

* The contents of this chapter was adapted from: Teixeira, C., Covas, J.A., Stützle, T. and Gaspar-Cunha, A., 2011. Engineering an efficient two-phase local search algorithm for the co-rotating twin-screw extruder configuration problem. *International Transactions in Operational Research*, 18 (2), 271-291.

4.1. Introduction

In the last decades, polymer systems with increasing performance levels have been developed, enabling the use of these materials in advanced applications. One way of achieving this involves mixing a new generation of additives with polymers in extruders. Co-rotating twin-screw extruders are very popular for this purpose due to their modular construction and consequent geometrical flexibility, as well as their excellent mixing capabilities.

As referred above, the performance of co-rotating twin-screw extruders for a given polymer compound depends on the operating conditions (e.g., feed rate, screw speed, and barrel temperature), material properties, and screw configuration. Manufacturers offer a variety of screw elements designed for conveying, mixing, kneading, devolatilizing, etc. This leaves the user with the problem of selecting specific screw elements from the set of available ones, and defining their location along the screw axis, such that the process performance is maximized. This problem is known as the twin-screw configuration problem (TSCP). In this work, a simplified version of the problem is considered, where the position along the screw axis of a number of known elements is to be determined. This version of the TSCP can be seen as a sequencing problem, where an optimal permutation of a given set of screw elements must be determined.

Typically, in the TSCP, two or more performance measures need to be considered simultaneously. Hence, the TSCP is actually a multi-objective combinatorial optimization problem [GAS05]. In practical situations, it is difficult, if not impossible, to define a priori exactly the weights for each objective, as the set of alternative solutions is unknown. Therefore, the multi-objective version of the TSCP is tackled in terms of Pareto optimization, i.e., the goal is to provide an approximation to the Pareto optimal set that is as good as possible. An additional difficulty of the TSCP is the evaluation of a sequence. Measuring the performance of a screw configuration through physical experiments is too costly and time consuming. Instead, an elaborate numerical simulation of the polymer flow is used to estimate the performance for a given screw configuration. This simulation is time consuming and one execution requires several minutes on the computing platform used, which are AMD Opteront 2216 dual-core processors running at 2.4 GHz with 2 MB L2-Cache. The high cost of evaluating a sequence of screw elements also strongly limits the number of sequences that can be evaluated in a feasible time and, consequently, a further goal of algorithmic approaches to the TSCP consists in providing high-quality approximations to the Pareto front with as few evaluations as possible.

In the present work, a stochastic local search (SLS) algorithm for various bi-objective TSCPs is presented. A bottom-up engineering approach is followed to develop the final bi-objective SLS algorithm. In particular, the following four steps are considered. First, different local search operators are applied to various single-objective versions of the TSCP. Next, different search strategies are tested in order to further improve the solution quality. In a third step, neighborhood restrictions are tested in order to reduce the number of sequence evaluations necessary to reach high-quality solutions. In a final step, the iterative improvement algorithms are extended to the bi-objective cases considered, integrating them into a variation of the two-phase local search (TPLS) framework [PAQ03]. The choice of the TPLS framework is induced by the high performance reached by previous algorithms [PAQ03, PAQ05]. The purely sequential SLS algorithm engineering process is motivated by the fact that the sequence evaluation is very time consuming, which also limits the number of experiments to perform in alternative design choices. Despite the potential disadvantages of such a sequential methodology, the final high performance of the developed TPLS algorithm justifies this choice. In fact, the comparison of TPLS algorithm with a multi-objective evolutionary algorithm (MOEA) [GAS04], which has been previously developed specifically for this problem, shows the very good performance of the final TPLS algorithm: it outperforms the evolutionary algorithm (EA) on some test instances and, more importantly, for a small number of candidate solutions' evaluations, it reaches much better approximations to the Pareto front.

This chapter is structured as follows: the TSCP is described in Section 4.2. The development of efficient single-objective iterative improvement algorithms is detailed in Section 4.3 and the experimental results with the multi-objective SLS algorithms and a comparison of their performance with the MOEA are presented in Section 4.4. Section 4.5 contains the major conclusions.

4.2. Extrusion Problem

4.2.1. Assembly and Performance of Screws

The screws of co-rotating twin screw extruders are generally assembled using three types of individual modules: i) conveying elements; ii) left-handed elements and iii) kneading blocks. The typical screw elements are identified in Figure 4.1 and the detailed description of each element is presented in subsection 3.2.1.

As left-handed elements and kneading blocks with neutral and negative staggering angles create a restriction to the flow (generating pressure), they are called restrictive elements. As shown in Figure 4.2, in practice, screw profiles contain a series of conveying elements with different geometrical characteristics (e.g., to induce material compression) separated by a series of kneading blocks and/or left-handed elements, in order to melt, mix, or seal (for devolatilization purposes) the flow (for more details on such elements and twin-screw extruders see [RAU86, WHI10]).

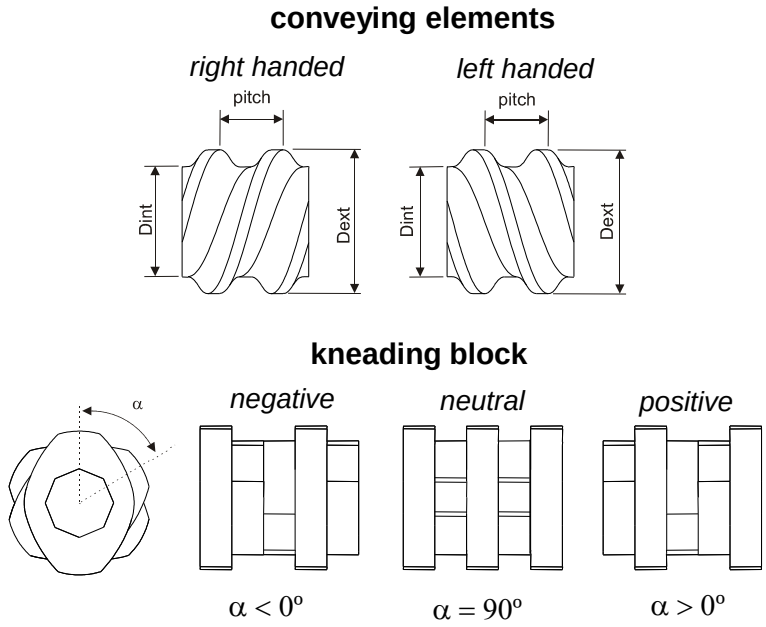

Figure 4.1. Typical screw elements.

Figure 4.2. Screw configuration (C: conveying element; KB: kneading block; LH: left handed).

The performance of a given screw configuration is evaluated through modeling of the polymer flow along the machine, i.e., predicting the axial evolution of the average and cumulative residence times, pressure, temperature, power consumption, average strain, specific mechanical energy (SME), viscosity, and channel degree of fill. The sequence, nature, and intensity of the individual process steps will depend on the requirements of the compound to be prepared. The simulation software developed previously [TEI07] comprises the various process steps developing along the extruder from hopper to die. As referred before, typically, the sequence of process steps is the following: (i) solids conveying without pressure; (ii) solids conveying under pressure; (iii) delay (formation of a melt film between the solids and the barrel); (iv) melting with a low liquid content; (v) melting with a low solids content; (vi) melt conveying under pressure; and (vii) melt conveying without pressure [TEI07].

Figure 3.5 illustrates the corresponding physical models. The respective corresponding individual process steps are described in detail in subsection 3.2.2.

In this type of extruders, the operator can set independently the output (i.e., the rate of feeding), the screw speed, and the temperature profile. Figure 3.9 shows the typical axial pressure and temperature profiles for a fixed screw geometry and operating conditions (for more details see subsection 3.4.1).

4.2.2. TSCP Instances

The computational experiments carried out in this work were performed using the features of a co-rotating twin-screw extruder Leistritz LSM 30-34 available at the University of Minho. In this machine, the screw is assembled by fixing 16 elements in a shaft. Four instances with different numbers of restrictive elements have been considered, as shown in Table 4.1. The instances differ in the number of restrictive elements considered.

Conveying elements with a length of 97.5 and 120 mm (and pitches of 45 and 30 mm, respectively) were always positioned upstream to guarantee enough initial conveying capacity. Thus, for the TSCP instances defined here, the goal is to determine the best permutation of 14 screw elements, which include transport elements with different lengths and pitches, kneading blocks with different staggering angles (-30°, -45°, and -60°), and one left-handed element. The

number of restrictive screw elements was progressively increased from one (instance TSCP1) to four (instance TSCP4).

Table 4.1. Screw elements used in each instance of the four case studies

Instance	Screw	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
TSCP1	Length	97.5	120	45	60	30	30	30	60	30	120	30	120	37.5	60	60	30
	Pitch	45	30	45	30	20	60	30	20	KB-60	30	30	60	20	45	30	20
TSCP2	Length	97.5	120	45	60	30	30	30	60	30	120	30	120	37.5	60	60	30
	Pitch	45	30	45	30	-20	60	30	20	KB-60	30	30	60	20	45	30	20
TSCP3	Length	97.5	120	45	60	30	30	30	60	30	120	30	120	37.5	60	60	30
	Pitch	45	30	KB -45	30	-20	60	30	20	KB-60	30	30	60	20	45	30	20
TSCP4	Length	97.5	120	45	60	30	30	30	60	30	120	30	120	37.5	60	60	30
	Pitch	45	30	KB -45	30	-20	60	30	20	KB-60	30	30	60	KB-30	45	30	20

Table 4.2 defines the objectives to consider, the direction of the optimization, and the range of variation of each objective. SME represents the mechanical energy required to move the screw per unit of material produced and is measured in MJ/kg. The viscous dissipation is calculated as the ratio between real and set temperatures. Hence, these two objectives should be minimized. Strain measures the product of the shear rate that is applied to the polymer by the time during which it is extruded and can be related to the degree of mixing, which should be maximized. Both viscous dissipation and strain are dimensionless.

Table 4.2. Optimization objectives, direction of optimization, and prescribed range of variation.

Objectives	**Direction**	**Xmin**	**Xmax**
Specific mechanical energy	Minimization	0.1	2
Viscous dissipation	Minimization	0.9	1.5
Average strain	Maximization	1000	15000

From both a single and a bi-objective perspective, this produces 12 instances, resulting from crossing the four instances TSCP1 to TSCP4 with the three possible objective combinations (three single objectives and, consequently, three pairwise combinations of two of the three objectives).
As mentioned above, the objective values are determined using a complex simulation program [TEI07]. The computation times that are required to evaluate one single screw configuration range between 1 and 3 min (measured on AMD Opteront 2216 dual-core processors running at 2.4 GHz

with 2 MB L2-Cache), depending mainly on the number of restrictive elements considered for an instance. As modifications to a screw configuration can have an effect on what happens before each screw element, even a slight modification requires a full re-simulation of the screw performance.

4.3. Single Objective Optimization

The main working hypothesis followed in this study is that effective single-objective SLS algorithms can be extended using the TPLS framework to yield high-performing algorithms for multi-objective problems [PAQ03, PAQ05]. Given the high cost of evaluating a candidate solution, this study is focus on the greedy improvement of solutions. In particular, on iterative improvement algorithms, which only accept better neighboring candidate solutions and terminate as soon as no improving neighboring solution is found [see Algorithm 4.1 for a high-level algorithmic outline of an iterative improvement algorithm; $g(.)$ is the evaluation function].

Algorithm 4.1. Iterative improvement local search (minimization)

1: s is an initial candidate solution
2: **while** s is not a local optimum **do**
3: s' is a neighbor of s such that $g(s') < g(s)$
4: $s := s'$
5: **return** s

Because the TSCP has not been tackled previously by local search algorithms, an appropriate neighborhood structure (step 1) is first determined. Then, several ways to reduce the number of candidate solutions evaluated by reconsidering different pivoting rules (step 2) and the use of neighborhood restrictions (step 3) are explored.

4.3.1. STEP 1: neighborhoods and pivoting rule

Of crucial importance for the performance of iterative improvement algorithms [HOO04] are (i) the choice of the neighborhood structure, as it defines the solutions that can be reached in one step by the local search, and (ii) the choice of the pivoting rule [PAP82], which determines the neighbor solution that replaces the current one. The two most used pivoting rules are considered [PAP82, HOO04]. The *best improvement* rule (often also called the steepest descent) selects among all the

neighboring solutions one that improves most the objective function; the *first improvement* rule searches the neighborhood in a specific order and accepts as the new candidate solution the first neighboring one that is better than the current.

A common way to define the neighborhood of a solution is to consider all the remaining solutions as being neighbors of a current solution that can be reached by applying a specific operator. For the TSCP, the adjacent swap, insert forward, insert backward, 2-swap and 2-exchange operators [HO004] have been considered. These operators apply the following types of moves to a current permutation $\pi = (\pi_1 ... \pi_i ... \pi_n)$, which represents the sequence of screw elements.

In *insert forward*, a screw element at position i is removed and inserted into a position $j > i$; the *insert backward* operator moves a screw element backward from a position i to a position j, with $j < i$. In both operators, the screw elements between positions i and j are shifted by one position to obtain again a correct permutation. A *two-swap* move swaps two screw elements at positions i, j with $i \neq j$. In an *adjacent swap*, the two elements that are swapped need to be in contiguous positions, i.e., in positions i and $i+1$. Finally, a *two-exchange* operator is also considered, which reverses the order of the screw elements between two positions i and j.

For each of these operators, 50 repetitions of an iterative improvement algorithm have been carried out on instance TSCP4, each repetition starting from a random initial permutation of the screw elements. Figure 4.3 shows boxplots of the performance of the different operators for instance TSCP4 in terms of the solution quality and of the number of evaluations for the maximization of the average strain, minimization of SME, and viscous dissipation. The best average quality is always obtained with the two-swap operator, independently of the pivoting rule used. For some objectives, the differences from the other operators are rather substantial. The statistical significance of the observed differences is confirmed by student´s $t-$test using Bonferroni corrections for multiple testing as reported in Table A.1 – Appendix A. (Statistical tests have been applied for each pivoting rule separately; an α level of 0.05 was used for the null hypothesis of equal performance of all operators.) Limited experiments using 10 repetitions of the iterative improvement algorithms on the other instances showed that the same trends with respect to the relative performance of the operators hold in the other instances.

The pivoting rules have a strong influence on the number of evaluations taken by the iterative improvement algorithms, the use of the first-improvement rule typically requiring only about half the

number of evaluations for reaching a local optimum when compared with the best improvement. The iterative improvement algorithm based on the adjacent swap neighborhood is the fastest, but it cannot be recommended due to the poor solution quality reached. Thus, the use of the two-swap operator and of the first-improvement strategy seems to be a good compromise.

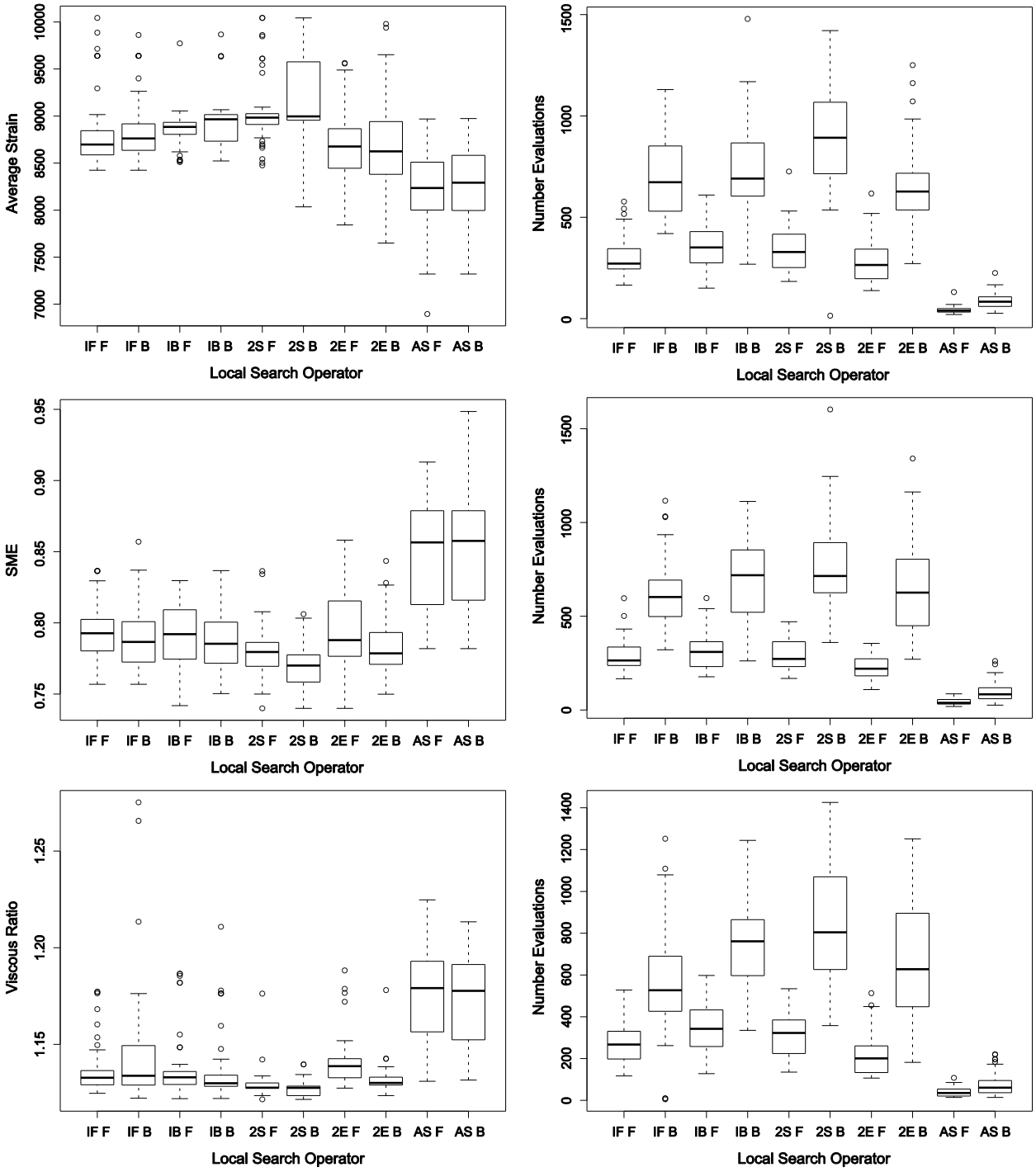

Figure 4.3. Performance obtained for the different operators studied for instance TSCP4 concerning the objective function (maximization of average strain, minimization of SME, and minimization of viscous dissipation, respectively), and the number of evaluations required. The boxplots are identified by operator and pivoting rule; the labels used are IF (insert forward), IB (insert backward), 2S (two-swap), 2E (two-exchange), and AS (adjacent swap) for the operators and B (best improvement) and F (first improvement) for the pivoting rules.

4.3.2. STEP 2: another look at the pivoting rule

According to the TSCP experts, the quality of a solution is strongly determined by the locations of the restrictive elements [RAU86, GAS05, WHI10]. Hence, instead of the standard lexicographic order of scanning the neighborhood, which is typically realized by two nested for loops, a different way of exploring the neighborhood is investigated by using a rule that first considers swaps where at least one restrictive element is involved. Only when no improved solutions can be found among this restricted neighborhood do, other swaps exclusively among non-restrictive elements is examined. To analyze the restricted neighborhood, where only swaps of restrictive elements are assumed, a first- and a best-improvement variant is also considered. In the *restrictive first* variant, we immediately move if an improvement is found. In the *restrictive best* variant, we first take a restrictive element e_r and then explore all possible swaps involving e_r if several improvements were found in this scan, we apply the best among those. Once improvements in the restricted neighborhood can no longer be found, the first improvement strategy that was described in the previous section is considered.

Table 4.3 shows the average and the standard deviation of the objective function values and the average number of evaluations obtained with the application of the three different search strategies (*sequential* refers to the first-improvement strategy from the previous section). The statistics are obtained across 10 independent runs of each algorithm, each run starting from a random initial permutation of the screw elements. In general, the restrictive best strategy yields the best average results in 10 of the 12 instances. In fact, for all objectives on TSCP1, TSCP2, and TSCP3, the best average results are obtained by the restrictive best. In some cases, particularly when the number of restrictive screw elements is small, the number of evaluations is also lower than that required by the sequential strategy. Given the better average performance of the restrictive best strategy when compared with restrictive first, the statistical significance of the differences between restrictive best and sequential were examined. The latter was never found to be statistically significantly better than restrictive best, while the opposite occurred in two out of 12 cases (after adjusting the α-level for multiple comparisons – see Appendix A, Table A.2). Despite the fact that on single instances the statistical evidence in favor of restrictive best is not too strong, which may be mainly due to the small sample size, the restrictive best strategy was chosen due to its better performance in the majority of the instances.

Table 4.3. Experimental results for the different search strategies for the maximization of average strain, minimization of specific mechanical energy (SME) and minimization of viscous dissipation.

Instance	Criterion	Search Strategy	Objective function		Evaluations	
			Mean	s.d.	Mean	s.d.
TSCP1	Average strain	Sequential	3320.52	108.10	293	65.59
		Restrictive first	3387.96	117.91	220.4	39.26
		Restrictive best	**3405.99**	111.75	211	30.34
	SME	Sequential	0.200981	0.00169	238.3	51.57
		Restrictive first	0.201519	0.00142	258.7	82.83
		Restrictive best	**0.200743**	0.00172	223.9	54.36
	Viscous dissipation	Sequential	1.0311	0.00064	155.7	37.39
		Restrictive first	1.0308	0.00061	173.5	42.95
		Restrictive best	**1.0306**	0.00079	236.7	108.66
TSCP2	Average Strain	Sequential	5110.99	74.21	246.9	73.49
		Restrictive first	5179.00	150.67	275.7	62.17
		Restrictive best	**5206.63**	110.66	256.7	62.49
	SME	Sequential	0.252564	0.01715	270.5	88.14
		Restrictive first	0.25975	0.01404	256	65.25
		Restrictive best	**0.252008**	0.018551	283	113.86
	Viscous dissipation	Sequential	1.1132	0.00364	137.8	57.23
		Restrictive first	**1.1122**	0.00427	146.8	67.97
		Restrictive best	**1.1112**	0.00445	156.8	66.87
TSCP3	Average Strain	Sequential	7182.32	271.41	328.1	115.44
		Restrictive first	7161.24	221.29	309.5	77.72
		Restrictive best	**7273.05**	175.95	318.8	70.79
	SME	Sequential	0.50613	0.03023	267.2	88.37
		Restrictive first	0.48585	0.01195	314.2	53.48
		Restrictive best	**0.48004**	0.00650	336	77.59
	Viscous dissipation	Sequential	1.12789	0.00164	255.9	75.35
		Restrictive first	1.12496	0.00343	263.5	83.30
		Restrictive best	**1.12472**	0.00277	320	105.62
TSCP4	Average Strain	Sequential	**9113.91**	353.58	351	176.52
		Restrictive first	9019.42	240.86	355.5	70.66
		Restrictive best	9065.36	418.68	349.2	77.83
	SME	Sequential	**0.77484**	0.01101	289.9	76.13
		Restrictive first	0.78058	0.01618	298.4	100.74
		Restrictive best	0.78481	0.01585	290.1	53.70
	Viscous dissipation	Sequential	**1.12799**	0.00183	355.9	89.42
		Restrictive first	1.12895	0.00174	365.8	134.74
		Restrictive best	**1.12799**	0.00112	289.4	97.36

The best average results for each instance and objective combination are indicated in bold face.

4.3.3. STEP 3: neighborhood restrictions

Given the significant computational time required by the modeling routine, it is useful to consider mechanisms targeted towards reducing the number of evaluations taken for the iterative improvement algorithms to be completed. The following neighborhood restrictions were considered, which can be applied either alone or in combination [HO004].

Restriction on the number of neighborhood scans: This consists in limiting the number of times that the full neighborhood is explored. The local search stops after a number of full neighborhood scans, even if further improvements may be possible – this may be the case if a solution has not yet been identified to be a local optimum.

Restriction on the indices: This neighborhood restriction limits the difference of the permutation indices for which a swap is considered (similar restrictions have been explored in [ADE92] for a scheduling problem). For the two-swap operator used here, we impose that for two screw elements at positions i and j, $j > i$, a swap of the two screw elements is considered only if $|j - i| \leq k$, where k is a parameter. The value of k for the number of neighborhood scans is adapted. In the first scan, the entire neighborhood is explored, while in subsequent scans, k is successively reduced. A schedule of 14, 10, 7, 7, . . . for the value of k is used, which was found to behave well in initial explorative trials with different schedules.

Do not look bits (DLB): In this strategy, a DLB is associated to each position. In the beginning, all DLBs are set to zero. If no improvement is found when scanning all neighboring solutions that involve moving an element at position i, then the DLB associated to position i is set to one and the position i is not considered. If in a subsequent search step one screw element (with a DLB equal to one) is swapped with another, its DLB is reset to zero. At each search step, only screw elements with their DLB equal to zero are used as a center of the search for improving moves. This technique is well known from its application to the traveling salesman problem. More details can be found in [BEN92].

Four experiments were performed, where each of these neighborhood restrictions was applied either individually, or all simultaneously; each algorithm was run 10 times on every instance. The underlying iterative improvement algorithm uses the two-swap neighborhood and the restrictive best strategy. In Table 4.4, a 95% confidence intervals is given that bound the loss of solution quality

when comparing the iterative improvement versions with all and without any of the neighborhood reduction techniques. The latter corresponds to the restrictive best strategy from the previous section. When the combination of all neighborhood restrictions is applied, the number of evaluations decreases significantly, between 16% and 58%, depending on the objective and instance, and the worsening of the solution quality is minor. In fact, for what concerns solution quality, the largest differences arise in the minimization of SME, where, in the worst case, with a 95% confidence, we can bound the performance loss to be at most 8.3% (instance 2); however, in most other cases, the bounds on the performance loss are much smaller; for the maximization of the average strain and the minimization of the viscous dissipation, they are typically below 1.6% and 0.7%, respectively.

Table 4.4. 95% confidence intervals (95% CI) for the reduction of the solution quality, when comparing the iterative improvement algorithms with all and without any neighborhood restrictions for the four instances and the three objectives (maximization of the average strain, minimization of the specific mechanical energy, and minimization of viscous dissipation).

Instance	Avg. diff (%)	95% CI	%Evaluation reduction
Maximization of average strain			
TSCP1	-0.170	[-0.555, + 0.215]	16.3
TSCP2	-0.486	[-1.144, + 0.172]	26.2
TSCP3	-0.647	[-1.609, + 0.315]	33.1
TSCP4	-0.649	[-1.411, + 0.114]	37.7
Minimization of SME			
TSCP1	+0.996	[-1.318, + 3.310]	58.1
TSCP2	+2.741	[-2.832, + 8.314]	35.7
TSCP3	+0.827	[-0.480, + 2.134]	38.5
TSCP4	+0.681	[-0.385, + 1.746]	34.1
Minimization of viscous dissipation			
TSCP1	+0.255	[-0.176, + 0.686]	43.1
TSCP2	+0.008	[-0.006, + 0.023]	20.1
TSCP3	+0.188	[-0.040, + 0.415]	36.1
TSCP4	+0.071	[-0.024, + 0.167]	36.4

Based on these results, the use of all neighborhood restrictions for the iterative improvement algorithms to tackle the multi-objective TSCP versions is considered. For simplicity, this final iterative improvement algorithm is denoted as II-F; it is an iterative improvement algorithm in the two-swap neighborhood that uses the restrictive best strategy and all neighborhood reduction techniques mentioned in this section.

4.4. Multi-Objective Optimization

As a final step, the II-F was integrated into the TPLS framework [PAQ03] and its performance on the bi-objective versions of the problem was compared with a previously proposed MOEA [GAS04].

4.4.1. Two-Phase Local Search

The TPLS framework exploits the possibility of applying single-objective SLS algorithms to aggregations of the various objective functions into a single one, i.e., TPLS follows the scalarized acceptance criterion search model [PAQ03].

A weighted sum formulation is used after the range of each objective is linearly transformed into the range [0, 1] using the bounds on the objective values given in Table 4.2; these bounds impose physical limits that also restrict the range of possible objective values. For a given weight λ, the value w of a solution s with an objective function vector $f(s)=(y_1,y_2)$ is computed as:

$$w=\lambda y_1+(1-\lambda)y_2 \text{ s.t. } \lambda\in[0,1]\subset\Re \tag{4.1}$$

TPLS modifies the weight λ [and, hence, the weight vector $(\lambda, 1-\lambda)$ for each of the two objectives] iteratively during the search, to focus on different areas of the Pareto front. An algorithmic outline of TPLS is given in Algorithm 4.2. TPLS starts from a random initial solution that is improved by a stochastic local search algorithm (SLS_1) using a weight, λ, which is either one or zero, depending on the objective from which TPLS starts, and adds this solution to the archive (lines 1–3). Next, TPLS solves, possibly by a different SLS algorithm SLS_2, a sequence of scalarizations; in this sequence, each scalarization (defined by the weight λ) starts from an initial solution generated by a previous scalarization (lines 4–6). The solutions found by this process are added to the archive (line 7) and the search stops once some termination criterion such as a maximum number of evaluations is reached and then the archive is filtered.

When compared with the original TPLS algorithm [PAQ03], this implementation has two main differences. First, all non-dominated solutions that are encountered in the iterative improvement algorithms are stored into the archive, as some of the candidate solutions explored could be non-supported, non-dominated solutions; in Paquete and Stützle [PAQ03], only the final solutions for each scalarization were considered. Second, the follow sequence of weight vectors is used [(1,0); (0,1); (1/2, 1/2); (3/4, 1/4); (1/4, 3/4); (1/8, 7/8), (3/8, 5/8), . . .], while in Paquete and

Stützle [PAQ03] a minimal weight change is applied by considering the sequence of weight vectors defined as $[i/m,(m-i)/m]$, $i=0, 1, ..., m$. For each weight λ, a new scalarization considers each of the two solutions s_1 and s_2 returned by the two closest weights as initial solutions and the one that has a lower scalarized objective value for the new λ is chosen. By this change a better anytime performance [ZIL96, DUB10] is expected. In fact, the disadvantage of the weight-setting strategy in Paquete and Stützle [PAQ03] is that the number of scalarizations needs to be fixed in advance without knowing how many evaluations will be required to complete the full TPLS scan. With the modified sequence of weight vectors, TPLS can be stopped at any time, while still hoping for a reasonable coverage of the Pareto front [DUB10].

Algorithm 4.2. 2phase search strategy

1: s is a randomly generated solution
2: $s' = \text{SLS}_1(s)$ /* First phase */
3: Add s' to Archive
4: **for all** weight vectors λ **do**
5: $s = s'$
6: $s' = \text{SLS}_2(s,\lambda)$ /* Second phase */
7: Add s' to Archive
8: Filter Archive
9: **return** Archive

4.4.2. Multi-Objective EA

EAs are probably the most used heuristic technique to tackle multi-objective optimization problems. It is often argued that they are particularly suited for multi-objective problems because they work with a population of solutions that can evolve in the direction of the optimal Pareto front [DEB01]. In this work, a multi-objective EA named Reduced Pareto Set Genetic Algorithm (RPSGA) is used, which was developed previously by Gaspar-Cunha and Covas [GAS04].

The outline of the RPSGA is illustrated in Algorithm 4.3. Initially, an empty external population p_e and an empty archive are formed (line 1) and an internal population p_i of N candidate solutions is randomly generated (line 2). At each generation, i.e., while a termination condition is not satisfied, the following operations are performed. First, the candidate solutions of p_i are evaluated by the

simulation routine (line 4). Next, a clustering technique is applied to reduce the number of solutions on the efficient frontier and to compute the fitness of each individual of p_i (line 5) and a fixed number of the best individuals are copied to p_e (line 6). If p_e is not full, individuals of p_i are selected for the application of the inver-over operator (lines 10, 11); if p_e is full, a clustering technique is applied to sort the individuals of p_e and a pre-defined number of the best individuals from p_e are incorporated into p_i and replace the lowest fitness individuals. All non-dominated solutions found during the computations are copied to the archive (line 12) and the algorithm returns finally all non-dominated solutions of the archive, which are obtained after filtering it. For the experiments developed, the parameter settings that have been recommended in Gaspar-Cunha and Covas [GAS04] are used. p_i and p_e have 100 and 200 individuals, respectively; a roulette wheel strategy was adopted for selection; and the probability for applying the inver-over operator is 0.8. More details and other parameter settings, for example, for the clustering technique, can be found in the original publication [GAS04].

Algorithm 4.3. RPSGA for TSCP

1: Initialize p_e (external population) and Archive to empty set
2: p_i is a randomly generated, initial population (internal)
3: **while** termination condition not satisfied **do**
4: Evaluate p_i
5: Evaluate individuals' fitness considering clustering
6: Copy best individuals to p_e
7: **if** external population full **then**
8: $p_e \leftarrow$ Clustering (p_e)
9: Copy best individuals of p_e to p_i
10: Select individuals for reproduction
11: Apply Inver-over operator to selected pairs of individuals
12: Add non-dominated solutions to Archive
13: Filter Archive
14: **return** Archive

4.4.3. Experimental Results

As a final step, the performance of TPLS algorithm is compared with RPSGA for various values of sequence evaluations. Owing to the difficulty in comparing the results of multi-objective algorithms [ZIT03, KNO06], a methodology based on attainment functions [FON96] is used. The attainment function attributes to each objective vector z, a probability that this point is attained in one single run [FON96]. The attainment functions of TPLS and RPSGA are estimated by running the algorithms several times and storing every time the set of non-dominated solutions obtained. The result is an empirical attainment function (EAF) [GRU01]. The differences in the EAFs between the two algorithms can be visualized by plotting the points in the objective space where the EAFs jump between different empirical frequencies and where the differences between the EAFs of the two algorithms are beyond a threshold [LOP06, LOP10a]. For an example, see the top plot of Figure 4.4. On the right side, the coordinates where TPLS attains points with a (much) higher frequency than RPSGA are indicated; the value of the difference is encoded in gray scale: the darker the points, the stronger the observed differences. The two extreme lines indicated in the plot connect the best points found by the two algorithms (grand best) and the points dominated in any run (grand worst); the line in the middle corresponds to the boundary of the region that is obtained in 50% of the runs of each algorithm, i.e., it represents the median attainment function surface (on the right side for TPLS and on the left side for MOEA).

The EAFs of each algorithm are estimated by running them 10 times for a maximum number of 3000 screw configuration evaluations. As we are particularly interested in a fast approximation to the Pareto front, the results obtained by TPLS and RPSGA are compared after 1000, 2000, and 3000 evaluations for each bi-objective instance; the comparisons are made in Figures 4.4–4.7 (TSCP1–TSCP4) for the bi-objective problem of minimizing SME and maximizing the average strain. The results on the instances TSCP1–TSCP4 with other objective combinations are qualitatively similar to those presented here and can be found in Appendix A.

In general, the experimental results show an advantage of TPLS over RPSGA, especially for 1000 evaluations. In fact, in this case, TPLS has strong differences in its favor (as indicated by the gray-scale codification of the differences); the size of objective space areas with advantages in favor of TPLS also increases as we go from TSCP1 (Figure 4.4) with one restrictive element to TSCP4 (Figure 4.7) with four restrictive elements. The examples with the clearest differences in favor of TPLS are the instances with three or four restrictive elements. For example, in the TSCP4 case

(Figure 4.7) for 1000 evaluations no points in favor of the MOEA could be found. For a larger number of evaluations, MOEA catches up with the local search algorithm. In fact, for 3000 evaluations of screw configurations, MOEA obtains somewhat better results than TPLS on instances TSCP2 (Figure 4.5) and, to a lesser extent, on instances TSCP3 (Figure 4.6). For TSCP1 (Figure 4.4), TPLS is still better performing, while for TSCP4 (Figure 4.7), it is clearly better than MOEA. Considering the number of restrictive elements, for 1000 and 2000 evaluations, the advantages of TPLS are larger for more restrictive screw elements. However, the trend is not very strong and for a larger number of sequence evaluations (3000), it is no longer obvious.

As a general conclusion, TPLS is clearly competitive in relation to an earlier MOEA for this problem and that it generally performs better for a small number of sequence evaluations. These results are very relevant, especially because each evaluation of a screw configuration takes a computation time that is in the range of minutes on current computer hardware, at the time of writing this work.

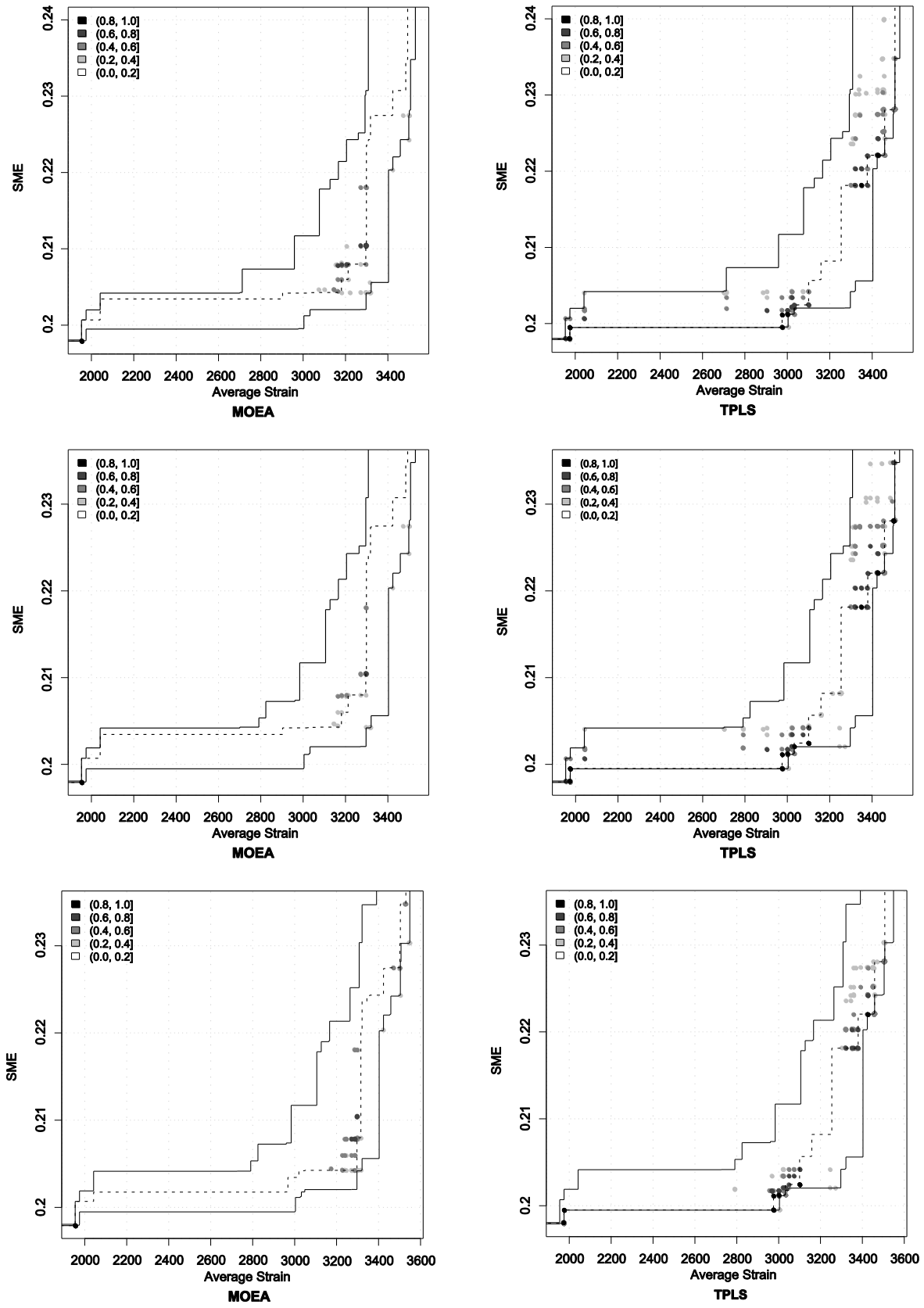

Figure 4.4. Results for instance TSCP1. Differences in terms of empirical attainment functions between the multi-objective evolutionary algorithm (MOEA) and two-phase local search (TPLS) after 1000 (top), 2000 (middle), and 3000 (bottom) evaluations of the simulation program for case study one. Advantages in favor of MOEA are indicated on the left side; those in favor of TPLS on the right side.

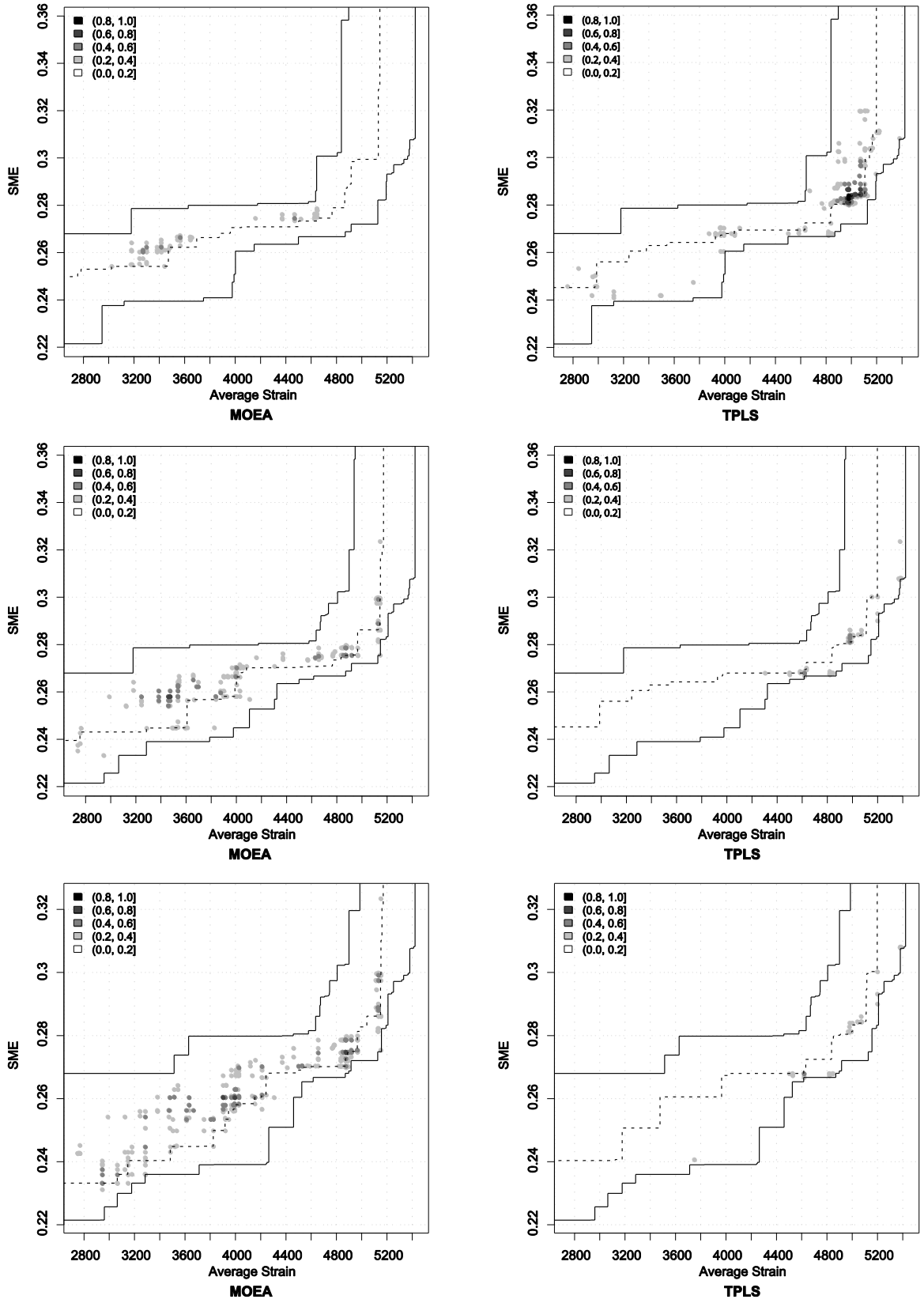

Figure 4.5. Results for instance TSCP2 (see the caption of Figure 4.4 for more details).

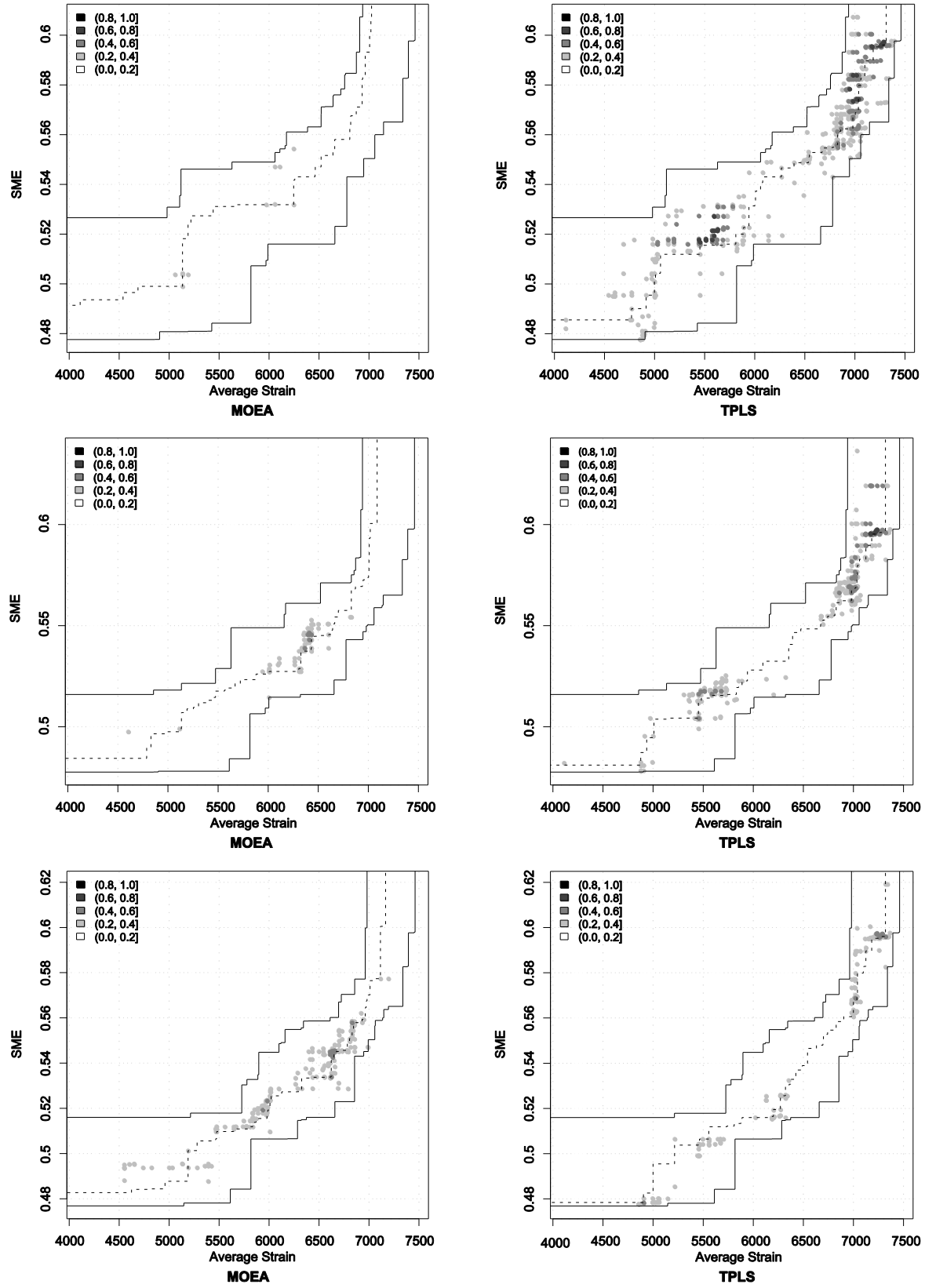

Figure 4.6. Results for instance TSCP3 (see the caption of Figure 4.4 for more details).

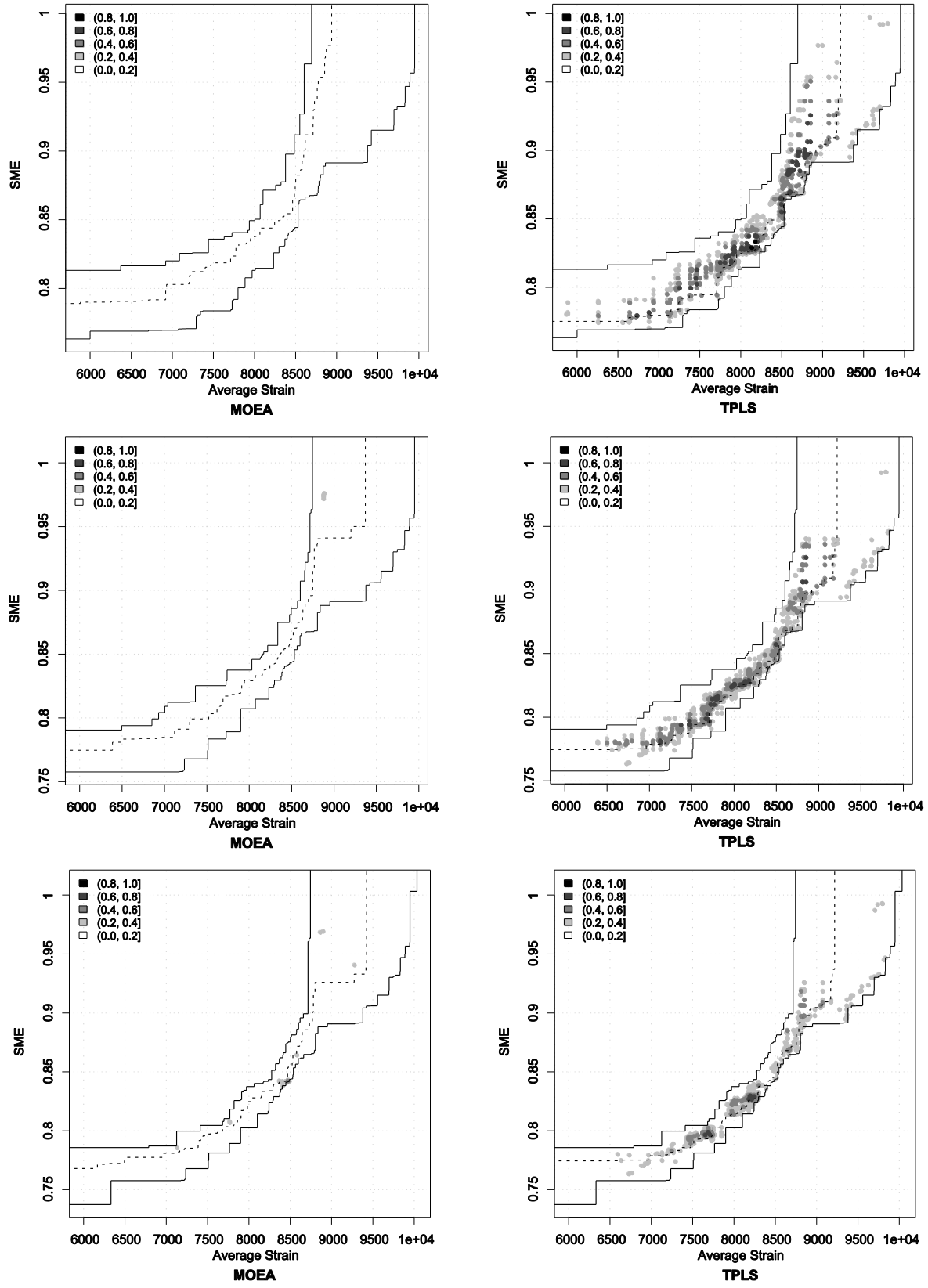

Figure 4.7. Results for instance TSCP4 (see the caption of Figure 4.4 for more details).

4.5. Conclusions

In this work, an effective TPLS algorithm for the bi-objective version of the TSCP was developed. Particular attention was paid to the definition of an appropriate neighborhood structure and to the exploitation of neighborhood restriction techniques. The single-objective algorithms were then extended to tackle bi-objective problems by integrating them into the TPLS algorithm framework. An experimental comparison of the TPLS algorithm to an MOEA, which was specifically designed for the TSCP, showed that TPLS is very competitive and that, especially for a low number of sequence evaluations, it reached better quality solutions than the MOEA.

There are a number of possibilities for further work. One is to consider hybrid algorithms. In a preliminary work, TPLS was used to seed a local search that then proceeds with a component-wise acceptance criterion (this algorithm is called the Pareto local search; [PAQ04, PAQ07]).

This hybridization had a positive impact on performance, but further experimental analysis with the new, more effective, iterative improvement algorithm is required. Another avenue for further research is the combination of TPLS with the RPSGA, or other MOEAs by, for example, seeding the MOEAs´ population with TPLS results, exploiting in this way TPLS´s fast convergence to a good approximation of the Pareto front. Finally, a main research interest is the extension of the algorithms to a variant of the TSCP: assuming there is a (large) set of screw elements and the task is to first select a smaller number of screw elements and then to sequence them. In this case, two related sub-problems arise, namely that of selecting the appropriate subset of screw elements and that of sequencing screw elements, making this problem a very challenging one.

Chapter 5

Pareto Local Search Algorithm*

A Pareto Local Search (PLS) algorithm was developed and applied to the screw configuration of co-rotating twin screw extruders (TSCP). This problem can be seen as a sequencing problem where a set of different screw elements are to be sequentially positioned along the screw in order to maximize the extruder performance. The results obtained were compared with previous results obtained with the Reduced Pareto Set Genetic Algorithm (a MOEA algorithm) and Two-Phase Local Search algorithm (TPLS) previously developed. These results show that the PLS algorithm, despite its conceptual simplicity, is able to generate screws with good performance.

---

* The contents of this chapter was adapted from: Teixeira, C., Covas, J.A., Stützle, T. and Gaspar-Cunha, A., 2010. Optimization of Co-rotating Twin-Screw Extruders Using Pareto Local Search. *In:* X. Gao, A. Gaspar-Cunha, M. Köppen, G. Schaefer and J. Wang, eds. *Soft Computing in Industrial Applications*. Vol. 75. Heidelberg: Springer, 3-10.

5.1. Introduction

In the previous chapter, an efficient single-objective iterative improvement algorithm was developed by studying different neighborhood structures, neighborhood search strategies and neighborhood restrictions. The aim was to develop an appropriate local search strategy to deal with the TSCP.

Basic iterative improvement algorithms can be adapted to deal with multi-objective problems considering two distinct strategies [TEG03]. One can take advantage of the good performance of SLS algorithms to solve single objective problems, by converting multi-objective problems into single problems that aggregate the various objectives into a single objective function – *scalar acceptance criterion (SAC)*. This was the strategy developed and tested in Chapter 4 with the Two Phase Local Search algorithm (TPLS) [PAQ03, TEI11]. Alternatively, successful single-objective algorithms can be readily extended to MOOPs adopting a *component-wise acceptance criterion (CWAC)*, where a new solution is accepted in the local search if it is non-dominated by any of the other previous solutions. Note that the difference between the two strategies is mainly based on the acceptance criterion considered on choosing a new solution that will replace the candidate solution.

Considering the good performance verified by the TPLS algorithm (see Chapter 4 for details), a SLS algorithm founded on CWAC strategy was also taken into account in this study. Several algorithms based on this approach can be found in literature such as Pareto Archived Evolution Strategy (PAES) [KNO99], Simple Evolutionary Multi-objective Optimizer (SEMO) [LAU02], Bicriteria Local Search [ANG04] and Pareto Local Search algorithm (PLS) [PAQ04, PAQ05, PAQ07].

Considering the good results obtained in the resolution of several combinatorial optimization problems, such as, bi-objective Traveling Salesman Problem [PAQ04] and bi-objective QAP [PAQ05], the PLS algorithm proposed by Paquete et al. [PAQ04, PAQ05, PAQ07] was adapted and applied to the TSCPs described in Section 4.2.2. For that, the iterative improvement algorithm II-F developed in chapter 4 was integrated into PLS framework.

The main ideas of PLS consist in the use of an archive, where all non-dominated solutions found so far are kept, and the exploration of the neighborhood of these solutions using non-dominance criteria to decide about the acceptance of new solutions.

The main objective of this chapter is the development of an efficient Pareto Local Search algorithm (PLS) to deal with the Twin Screw Configuration Problem (TSCP). Here, the simplicity of the PLS will be explored aiming to a future hybridization with others metaheuristics.

This chapter is organized as follows. In Section 5.2 the framework of the PLS algorithm is presented and described. Then, its performance is compared with TPLS and MOEA algorithms making use of EAFs. The agreement of the solutions with the physical process is presented and discussed in Section 5.3. Finally, Section 5.4 presents the major conclusions.

5.2. Pareto Local Search

Pareto Local Search (PLS) is based on the exploration of the neighborhood of one candidate solution and is the extension of the single-objective SLS algorithms to multi-objective problems by the application of the Component-Wise Acceptance Criterion (CWAC), i.e, the algorithm accepts neighboring solutions which are non-dominated solutions with respect of the current solution and all solutions in the archive [PAQ04, PAQ05].

A key component of any local search algorithm is the definition of which solutions are neighboring. Take into account the previously studies [TEI11] (Section 4.3), the called iterative improvement algorithm II-F was integrated into PLS framework.

The outline of the PLS algorithm is presented in Algorithm 5.1. It starts from a random initial solution that will set its visited flag to zero and will be saved in archive (lines 1-3). Then, its neighborhood is explored making use of the iterative improvement algorithm II-F. All non-dominated solutions, with respect of the current solution *s*, identified in the neighborhood exploration are added to the archive, if they are not dominated by any of the solutions in the archive, otherwise are eliminated. Considering that the application of this procedure can generate some dominated solutions, the archive is also updated by removing dominated solutions as result of adding a new solution into archive (lines 4-8). When all the neighborhood of the initial solution *s* is explored, its visited flag is set to one and a new solution is picked from the archive (lines 9-10). This search process is repeated until a stop criterion is reached such as number of evaluations or all solutions in the archive has been explored. Finally, the archive is filtered.

PLS has a natural stop criterion. However previous studies showed, in some cases, a too strong increase of the number of solutions in the archive and, consequently, the number of evaluations required to complete the entire search process is higher than the limit imposed for the algorithms previously developed.

Thus, in order to make a fair comparison between all algorithms under study, the PLS was stopped when 3000 evaluations were reached. To guarantee a complete search along the Pareto front and to avoid cycles an archive bounding technique proposed by Angel et al. was applied [ANG04]. This technique divides the objective space in a grid of hypercubes and allows only one non-dominated solution to occupy a given hypercube.

Algorithm 5.1. PLS algorithm

1: s is a randomly generated solution
2: visited = 0
3: Add s to Archive
4: **while** termination condition not satisfied **do**
5: **for all** $s' \in N(s)$ **do**
6: Evaluate(s')
7: **if *s'*** is not dominated **then**
8: UpdateArchive( )
9: s flag = 1
10: $s \leftarrow$ PickArchive( )
12: filter Archive
13: **return** Archive

5.3. Experimental Results

The PLS algorithm was tested using the individual screw elements presented in Table 4.1 for a laboratorial Leistritz LSM 30.34 co-rotating twin screw extruder. Four instances and three different case studies (total of twelve optimization cases) were considered as presented, respectively, in Tables 4.1 and 4.2. The same procedure applied for the other algorithms was considered, i.e., each optimization run was performed ten times using different seed values and each run was stopped when 3000 evaluations were reached.

An important factor to take into account in the optimization process is the physical coherence of the solutions obtained. Thus, in order to demonstrate the capacity of the PLS algorithm to deal with the TSCP, Figure 5.1 shows a Pareto front for TSCP4 and case study two (optimization of average strain and viscous dissipation) considering a single run. As expected, the viscous dissipation (to be minimized) increases with the average strain (to be maximized). The viscous dissipation (measured

as the ratio between the average melt temperature and the set barrel temperature) is smaller when the restrictive elements are separated by conveying elements, since in this case the increase in temperature is also smaller. The opposite is true for the case of the average strain.

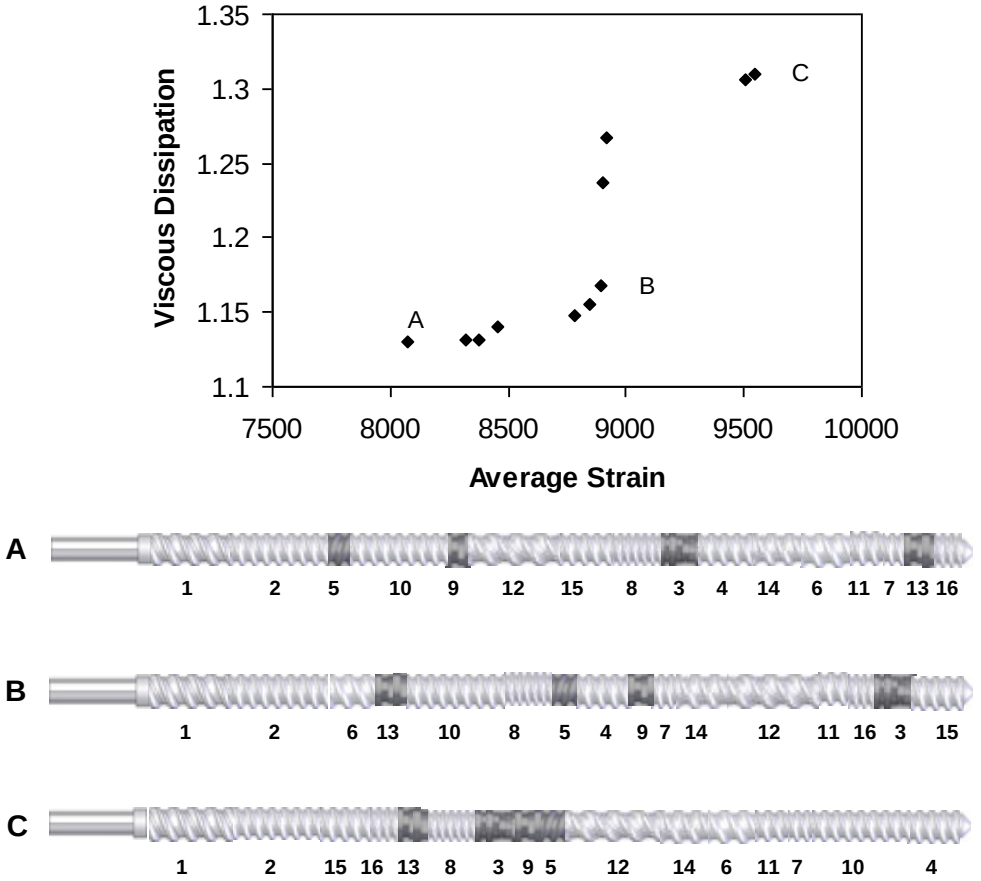

Figure 5.1. Pareto front for case study two, TSCP4 and run 1 and some screw configurations generated.

The results obtained with the PLS algorithm presented here were compared with the results obtained with the TPLS and MOEA algorithms (RPSGA) developed previously. Figures 5.2 to 5.5 show the comparison between PLS and TPLS algorithms in terms of EAFs for the twelve case studies (see Table 4.1 and 4.2 for details). For TSCP1 and TSCP2, where the number of restrictive screw elements is one and two, respectively, the differences between the two algorithms are not very significant (Figures 5.2 and 5.3). Some cases in favor of TPLS algorithm, others in favor of PLS. However, when the number of restrictive screw elements increases (TSCP3 and TSCP4 instances) the differences between the algorithms increase significantly in favor of PLS algorithm (Figures 5.4 and 5.5). When comparing the PLS algorithm with the RPSGA (Figures 5.6 to 5.9) two distinct behaviors can be observed. For the instances with a smaller number of restrictive screw

elements (TSCP1 and TSCP2) the results are slightly better for the RPSGA algorithm (Figures 5.6 and 5.7). One the other hand, PLS is significantly better than RPSGA for instances with a higher number of restrictive screw elements (Figures 5.8 and 5.9). These results show that PLS, despite its simplicity, can deal very well with the TSCP, mostly when the complexity of the problem increases (higher number of restrictive screw elements).

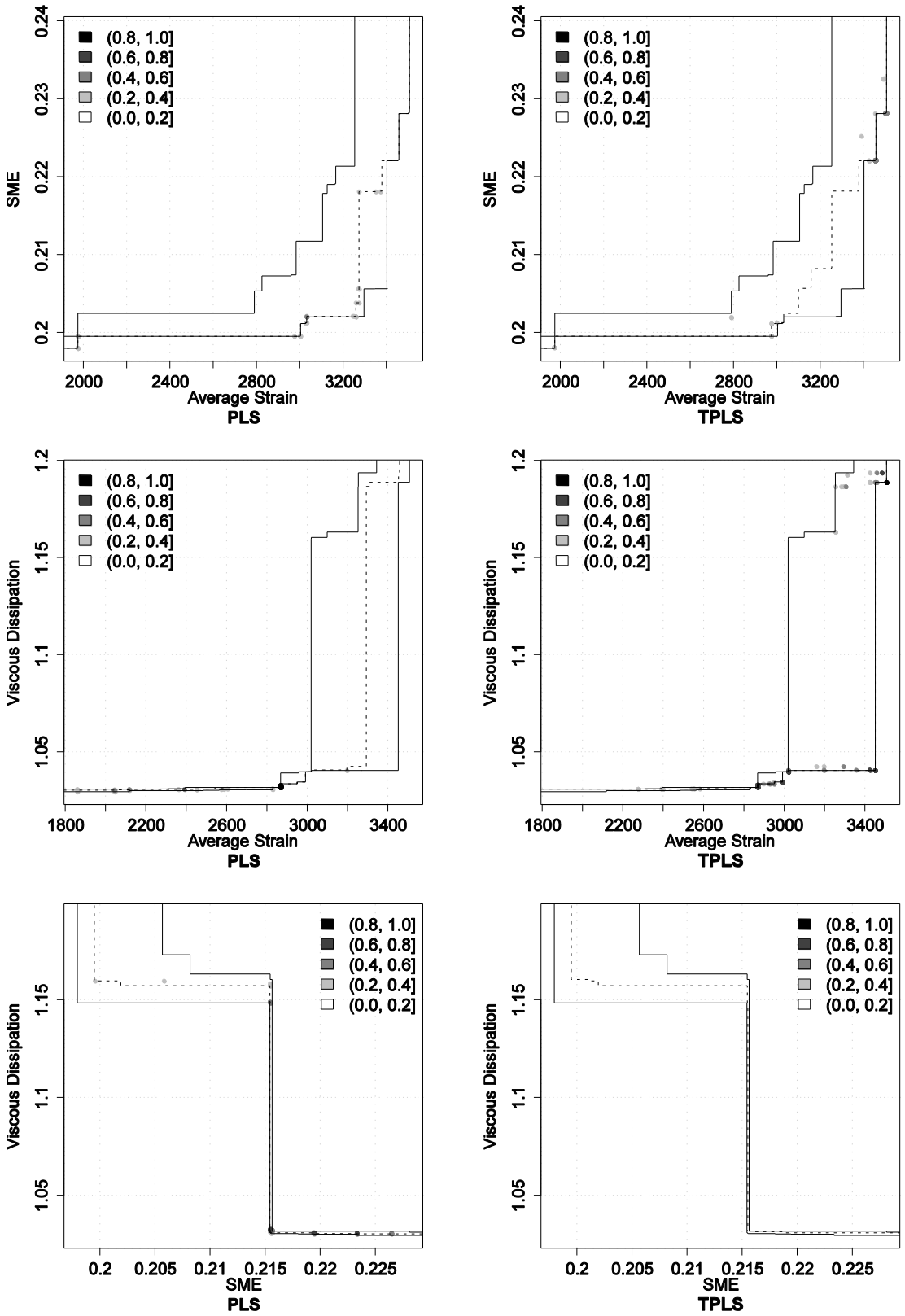

Figure 5.2. Results for instance TSCP1. Differences in terms of empirical attainment functions between the pareto local search (PLS) and two-phase local search (TPLS) for case study one (top), two (middle) and three (bottom) after 3000 evaluations. Advantages in favor of PLS are indicated on the left side; those in favor of TPLS on the right side.

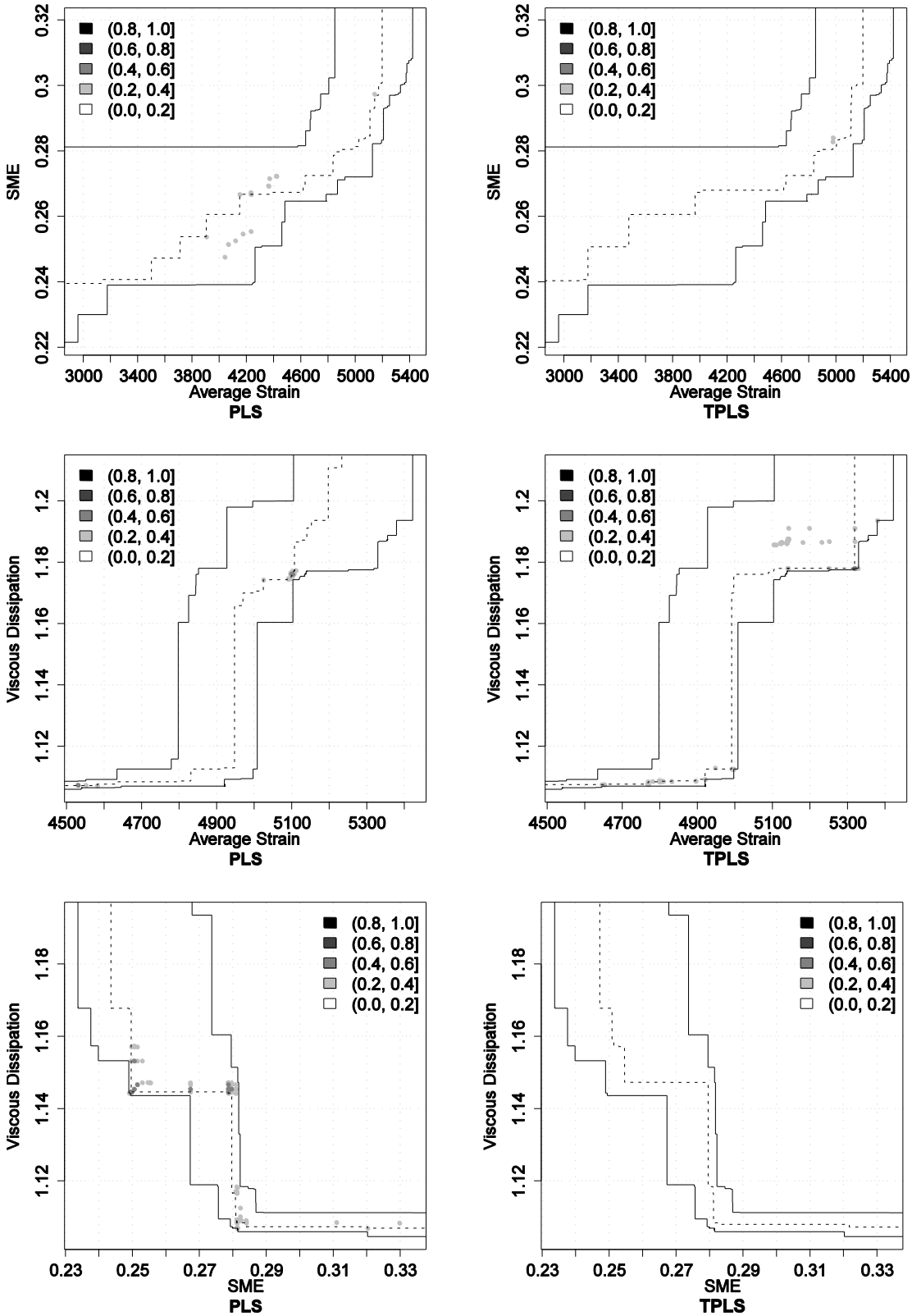

Figure 5.3. Results for instance TSCP2 (see the caption of Figure 5.2 for more details).

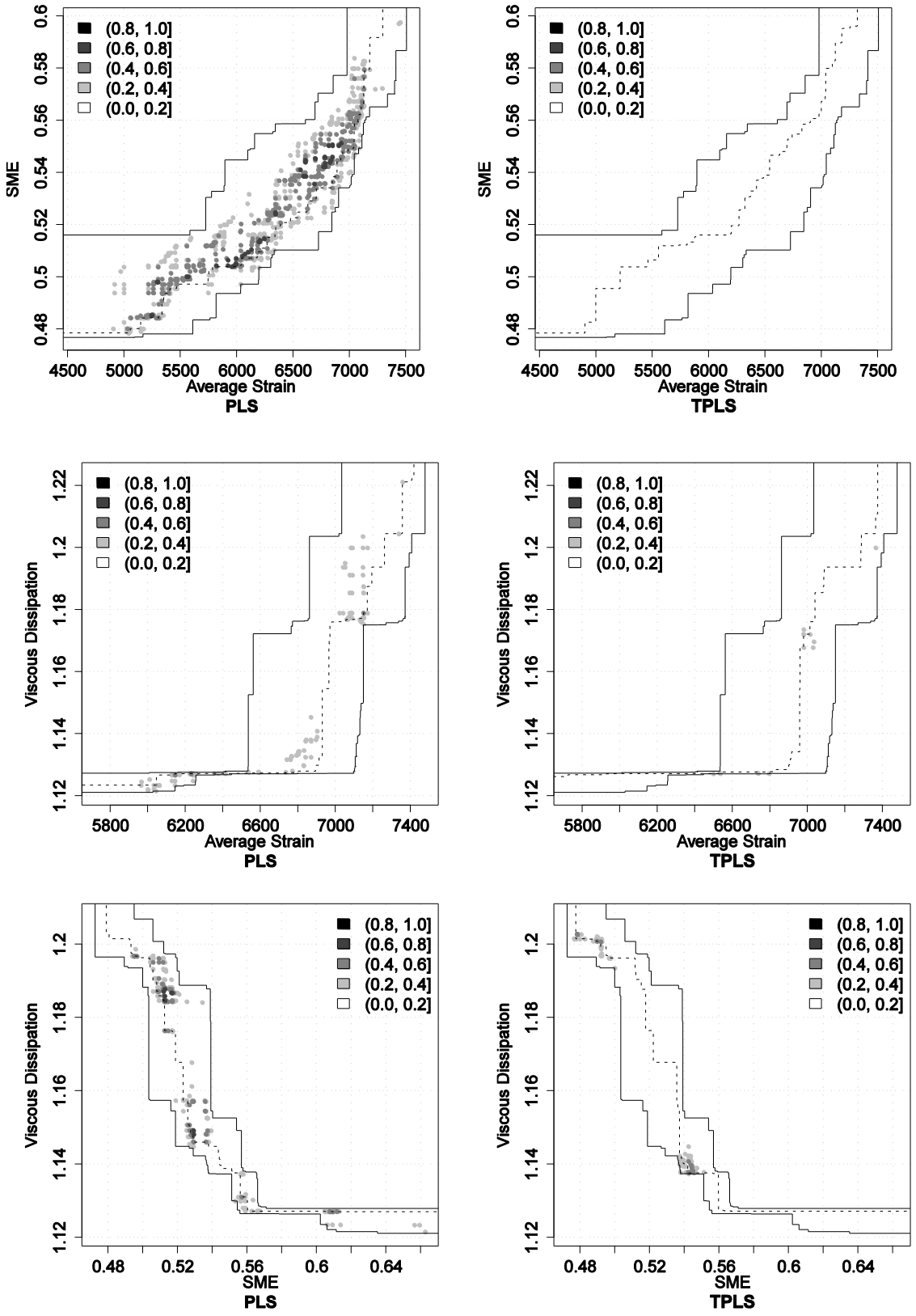

Figure 5.4. Results for instance TSCP3 (see the caption of Figure 5.2 for more details).

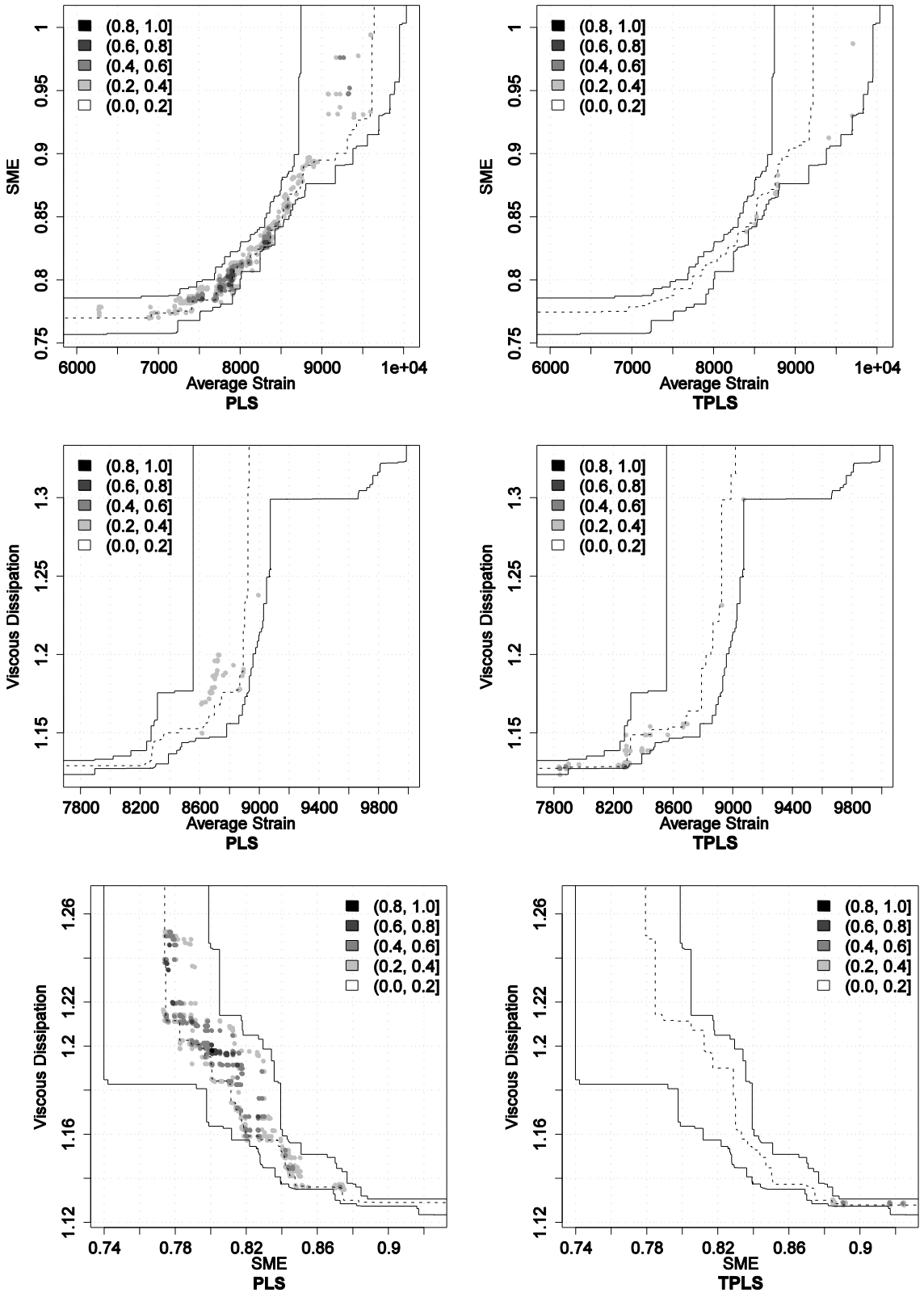

Figure 5.5. Results for instance TSCP4 (see the caption of Figure 5.2 for more details).

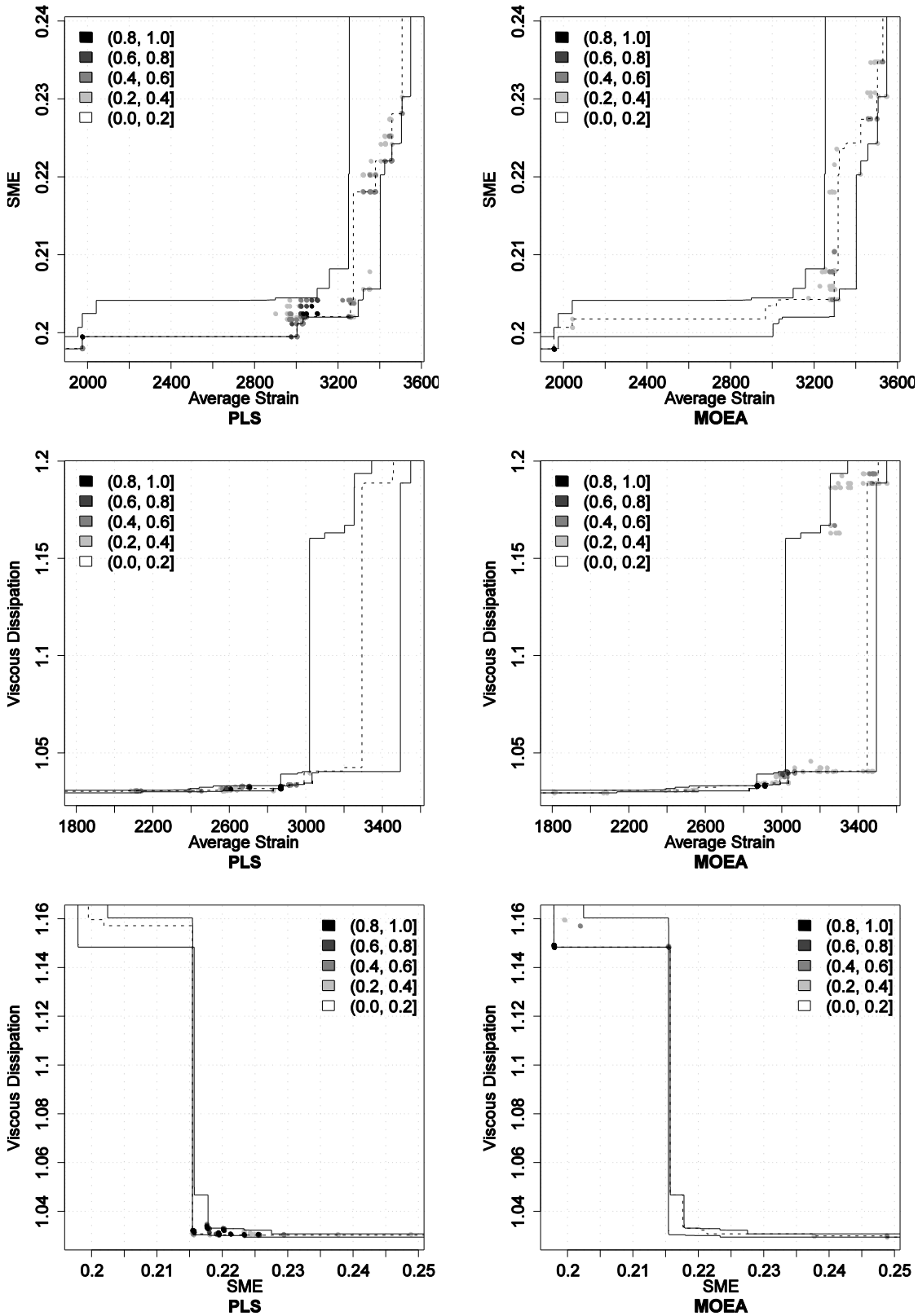

Figure 5.6. Results for instance TSCP1. Differences in terms of empirical attainment functions between the pareto local search (PLS) and multi-objective evolutionary algorithm (MOEA) for case study one (top), two (middle) and three (bottom) after 3000 evaluations. Advantages in favor of PLS are indicated on the left side; those in favor of MOEA on the right side.

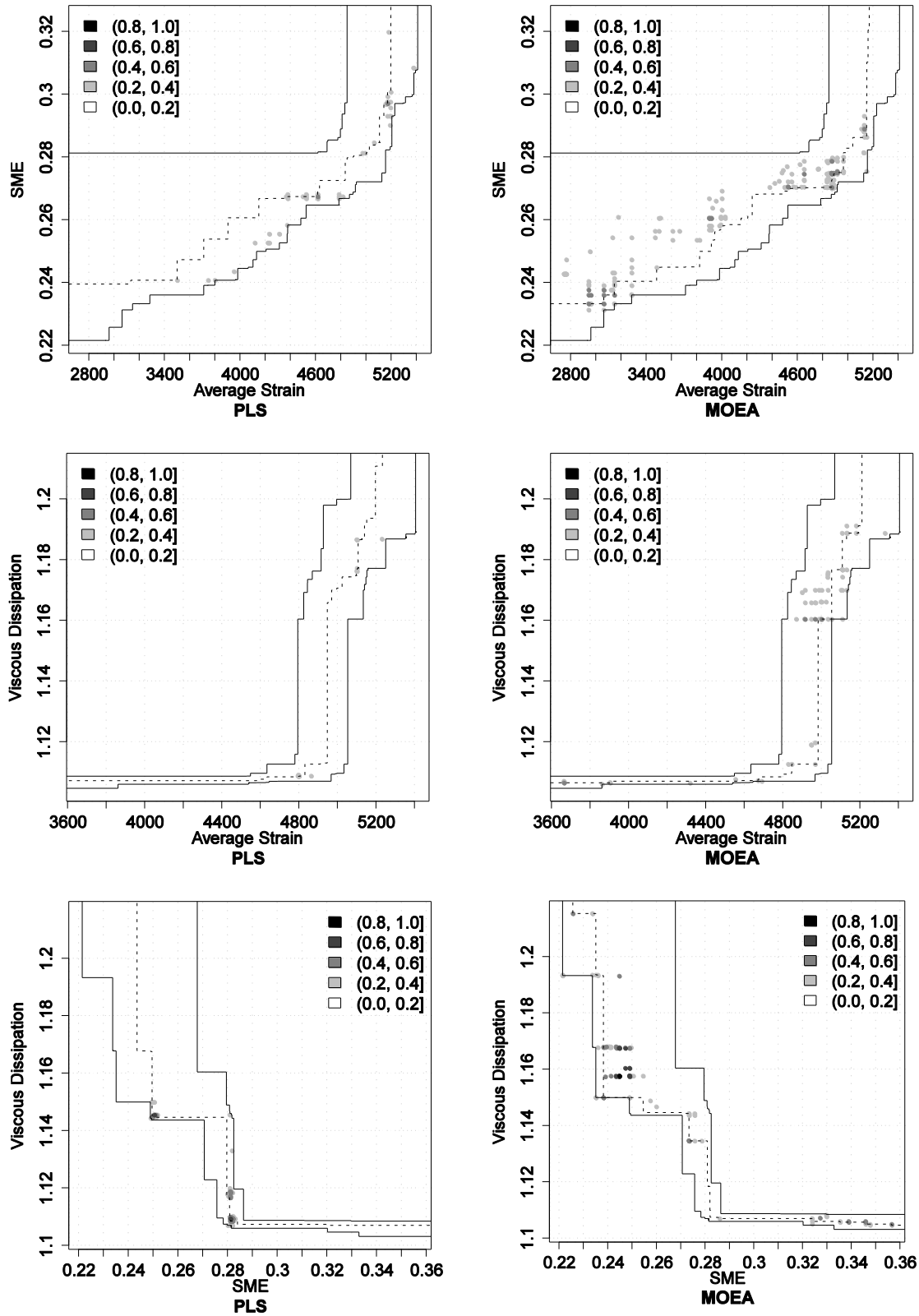

Figure 5.7. Results for instance TSCP2 (see the caption of Figure 5.6 for more details).

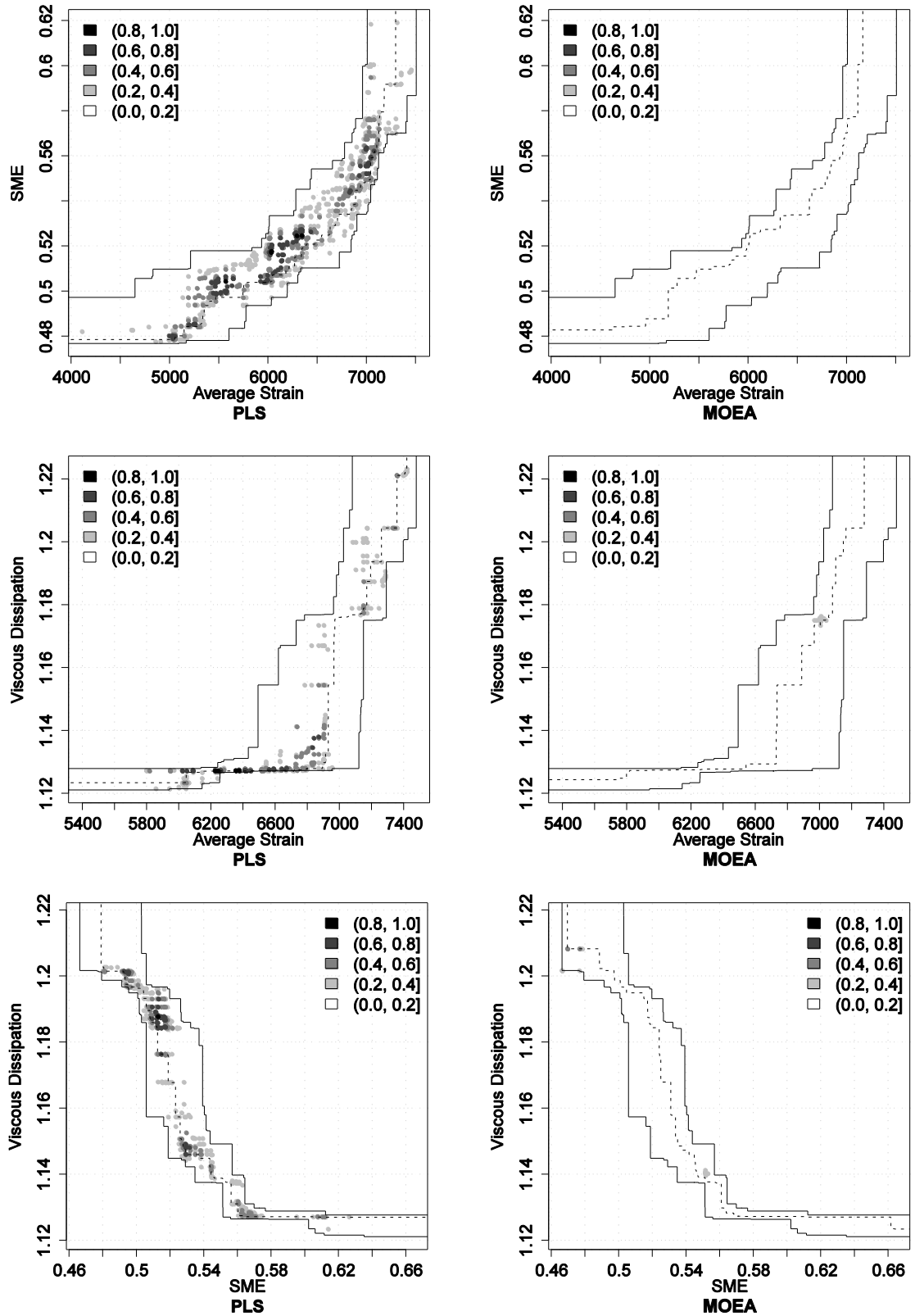

Figure 5.8. Results for instance TSCP3 (see the caption of Figure 5.6 for more details).

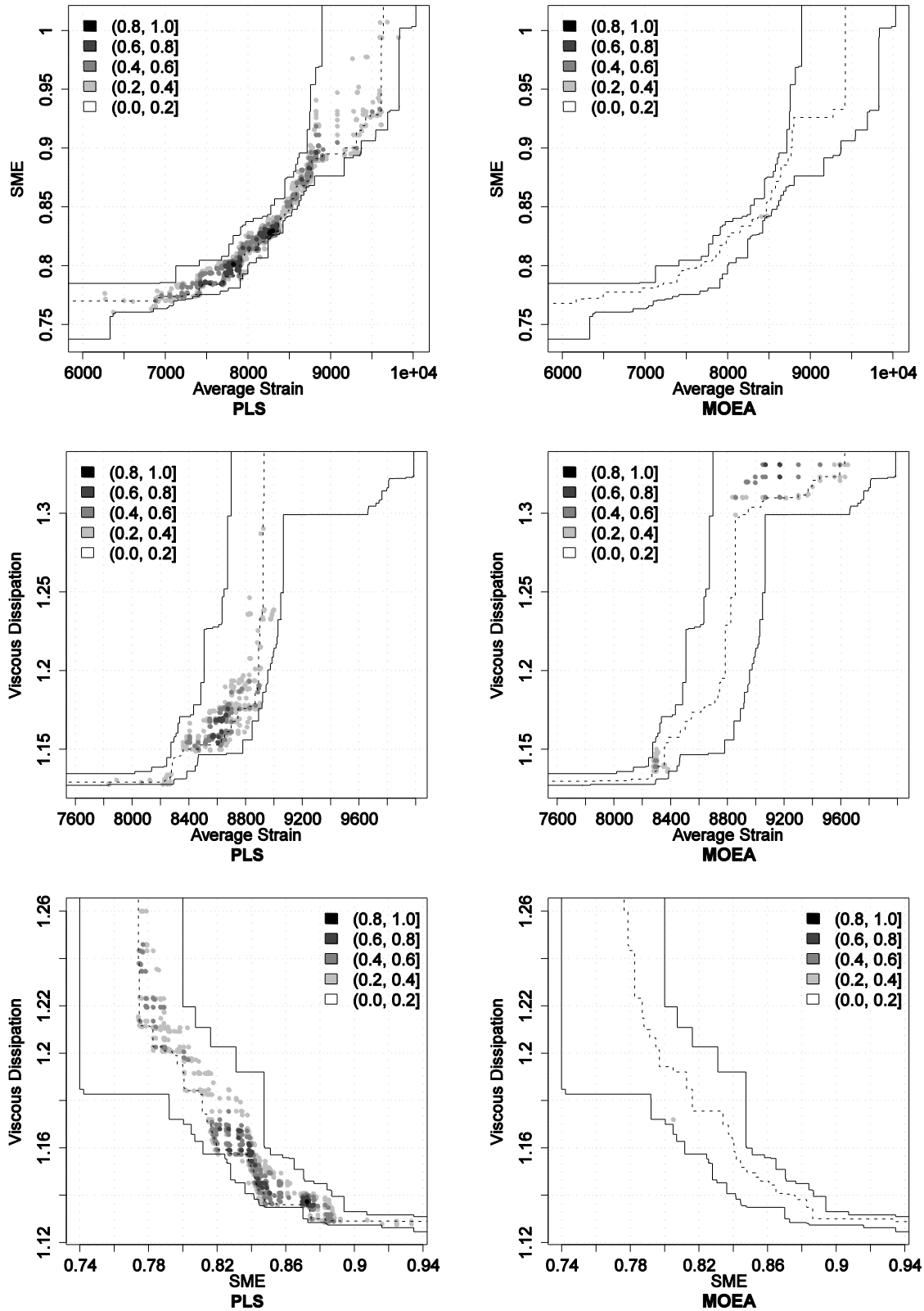

Figure 5.9. Results for instance TSCP4 (see the caption of Figure 5.6 for more details).

5.4. Conclusions

In this chapter, a Pareto Local Search algorithm was applied with success to the Twin Screw Configuration Problem. Take into account previous experiments, the called iterative improvement algorithm II-F was integrated into PLS framework and its performance was confronted with the knowledge about the extrusion process. The solutions obtained comply with the available scientific and technical knowledge on the process and have physical meaning.

Finally, the performance of PLS was compared with the RPSGA and TPLS algorithms using the EAF methodology. The good performance obtained with the PLS algorithm alone is somewhat surprising, mainly for instances with a higher number of restrictive screw elements, since it is a very simple method. In addition, the results indicate that the development of a hybrid algorithm through the incorporation of PLS, for example, as post processor, can be a good way to further improve the search process of others techniques, such as RPSGA and TPLS algorithms.

Chapter 6

Multi-Objective Ant Colony Optimization*

In this chapter, a multi-objective ant colony optimization (MOACO) algorithm was adapted to deal with the twin-screw configuration problem (TSCP). The influence of different parameters of the MOACO algorithm was studied and its performance was compared with that of a previously proposed multi-objective evolutionary algorithm (Reduced Pareto Set Genetic Algorithm – RPSGA) and a two-phase local search algorithm (TPLS). The experimental results showed that MOACO algorithms have a significant potential for solving the TSCP.

---

* The contents of this chapter was adapted from: Teixeira, C., Covas, J.A., Stützle, T. and Gaspar-Cunha, A., 2012. Multi-objective ant colony optimization for the twin-screw configuration problem. *Engineering Optimization,* 44 (3), 351-371.

6.1. Introduction

As referred before, assembling a screw from individual screw elements is an essential but challenging task in polymer industry, which might dictate a priori the success of the entire extrusion process. Extruder manufacturers offer a wide variety of geometries, with distinct conveying and mixing characteristics (for an overview of available screw elements, see [SAK91] and [KOH07]) which increase the complexity of the Twin Screw Configuration Problem (TSCP).

A first step attempted to solve the TSCP consisted in applying multi-objective extensions of local search algorithms. In particular, the use of iterative improvement algorithms embedded into the Two-Phase Local Search (TPLS) and Pareto Local Search (PLS) framework provided good results in comparison with RPSGA [TEI10b, TEI11] (see also chapters 4 and 5).

Ant Colony Optimization (ACO) is a relatively recent and powerful metaheuristic for tackling combinatorial problems. ACO takes inspiration from real ants' foraging behavior to define algorithmic solutions to computationally hard optimization problems. The first ACO algorithm (Ant System) was proposed by Dorigo et al. in the early nineties. Later followed the definition of the ACO metaheuristic framework by Dorigo and Di Caro [DOR99], which provides a high-level model for ACO algorithms. In ACO algorithms, ants exchange information indirectly by depositing pheromone that influences the choice of other ants. A recent overview on ACO algorithms and their applications can be found in [DOR10].

Several approaches have been proposed to apply ACO algorithms to Multi-Objective Optimization Problems (MOOP) such as multi-objective scheduling, vehicle routing and portfolio selection [MAR99b, IRE01, DOE04, GAR07, ANG09, LOP09, LOP10b]. A detailed review and comparison of the available multi-objective ACO (MOACO) algorithms for solving the bi-objective traveling salesman problem can be found in [GAR07]. A more recent review provides a detailed classification of MOACO algorithms [ANG09]. Several design alternatives of how to extend ACO algorithms to multi-objective optimization problems have been studied [LOP09] and the algorithmic design choices in available MOACO algorithms have been analyzed [LOP10b].

This chapter investigates MOACO algorithms as an alternative solution method for the TSCP. In particular, different design choices for MOACO algorithms are considered and their impact on MOACO's performance is analyzed for various bi-objective TSCP instances. Based on the insights gained from this study, a final MOACO algorithm for the TSCP is defined and its performance compared with RPSGA and TPLS algorithm. The computational results show that the proposed

algorithm results in approximations to the Pareto front that are better than those of RPSGA on several TSCP instances. This indicates that MOACO algorithms are very promising for tackling the TSCP.

This chapter is structured as follows. Section 6.2 presents the characteristics of the TSCP as well as the relevant aspects of the modeling routine. The multi-objective algorithms studied are described in Section 6.3 and the examples to be analyzed are presented in Section 6.4. Section 6.5 contains a discussion of the MOACO parameters and a comparison between MOACO, MOEA and TPLS results. Finally, the main conclusions are given in Section 6.6.

6.2. Twin Screw Configuration Problem

6.2.1. The Extrusion Problem

Figure 6.1 illustrates the typical layout of a co-rotating twin-screw extruder. As referred before, the machine contains two parallel identical screws that can rotate at constant (but tuneable) speed inside a heated barrel. The material enters upstream via a gravimetric or volumetric feeder set to work at a given rate assuring that the screws will work partially filled along most of their length. By action of the screw rotation, the material progresses axially with a complex flow pattern (it is repeatedly transferred between the channels of the two screws), while it is subjected to distinct thermomechanical environments (shear stresses, temperature, residence time), depending on the local screw geometrical features.

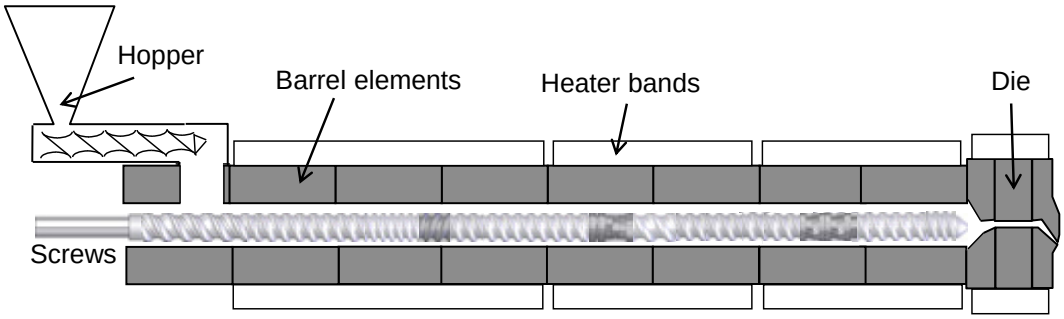

Figure 6.1. Co-rotating twin-screw extruder: example of machine layout.

As it progresses downstream, the material melts due to the combined effect of heat conduction from the barrel, heat generation due to friction and plastic deformation. Further downstream the melt may cross mixing zones that induce variable levels of distributive and/or dispersive mixing, flow along partially filled channels where new components - typically fillers or reinforcements - may be added, or devolatilization is attempted. A repetition of these operations along the screw is also possible. The sequence will depend on screw and barrel design, location and type of accessories and operating conditions. Towards the screw tips, the melt is pressurized and flows through the shaping die [WHI01b, WHI01c, WHI10, VER98, GAS02]. Screw speed and output can be controlled independently, although they both affect flow and heat transfer inside the machine.

As explained above, in most machines the screws are built by assembling a number of screw elements. These can vary in type (see Figure 4.1) and, for each, variations in pitch, helix or staggering angle, and length or number of kneading disks exist.

The technical features of each screw element type (conveying, left-handed elements and kneading blocks) are described in subsection 3.2.1. The restrictive character increases the local residence time and hydrodynamic stress levels, thus inducing polymer melting (if they correspond to the first restrictive section upstream) and/or distributive and dispersive mixing (tuneable via the staggering angle). Therefore, the screw profile of the extruder depicted in Figure 6.1 comprises one melting/mixing and two mixing zones, separated by four conveying sections. As the inlet material progresses along the screw, it quickly approaches the first restrictive zone, where the combination of pressure, temperature and local residence time will cause melting. If this zone is long enough, substantial dispersive and distributive mixing will develop (and, when chemical reactions are also involved, high chemical conversions may be reached). Melt flow during the second and third mixing zones will essentially contribute to better dispersive mixing. In fact, this type of mixing requires exposure to sufficiently high stresses during sufficient time. However, it is preferable to design a screw with several mixing zones instead of having one of equivalent length, as pressure and viscous dissipation in the latter would reach prohibitive levels. On the other hand, this design will generate complex pressure profiles, as flow develops along sequences of partially filled and fully filled channels.

6.2.2. Modeling Routine

Due to the complexity of the flow and heat transfer in twin screw extruders and the need to generate sufficiently accurate predictions of the process response upon changing its main parameters, a numerical modeling routine was developed [TEI07, TEI10]. For the purposes of this work, this routine can assumed to be a "black box". The inputs to this modeling routine are the material properties, the screw configuration (in the present case, the screw configuration is to be optimized) and the operating conditions (screw speed, barrel set temperature and mass output). The outputs are the parameters that characterize the process performance; these parameters are used as the objectives to be optimized.

It is clear that different screw configurations will create inherently different thermo-mechanical environments. The linkage between operating conditions, screw geometry/configuration, materials properties and the extruder performance is made through the modeling routine. Thus, the modeling routine must (i) be capable of describing accurately the relevant flow and heat transfer phenomena, (ii) be sensitive to changes in geometrical or operational parameters and (iii) require moderate computational resources. A detailed description of the routine is given elsewhere [TEI07, TEI10]. See also chapter 3 for more details. It encompasses all steps from material inlet to die exit, including solids conveying under or without pressure, melting, melt conveying under or without pressure and die flow. Figure 6.2 presents three different screw profiles that could be used in the extruder of Figure 6.1 and that are described in detail in Table 6.1. Each screw comprises 16 elements; right handed and left handed elements are identified by their length and pitch, while kneading blocks (KB) are defined by their length and staggering angle. In practice, at most the extrusion companies deal with 25-30 elements.

Figure 6.2. Examples of screw profiles to be used in the extruder of Figure 6.1.

Table 6.1. Geometrical identification of the screws illustrated in Figure 6.2.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	Length	97.5	120	45	60	30	30	30	60	30	120	30	120	37.5	60	60	30
	Pitch	45	30	KB -45	30	60	-20	30	20	KB -60	30	30	60	KB -30	45	30	20
B	Length	97.5	120	60	30	120	30	45	60	30	30	30	120	60	60	37.5	30
	Pitch	45	30	20	KB -60	30	30	KB -45	30	-20	60	30	60	45	30	KB -30	20
C	Length	97.5	120	45	30	60	30	120	37.5	60	60	30	30	30	30	60	120
	Pitch	45	30	KB -45	-20	30	30	60	KB -30	45	30	KB -60	20	60	30	20	30

Figure 6.3 and Table 6.2 demonstrate how the performance of these screws is distinct when using identical operating conditions (barrel and die set to 220°C, screws rotating at 150 rpm and feed rate of 8 kg/hr). A polypropylene polymer (ISPLEN PP 030G1E from Repsol) is being processed, having physical, thermal and rheological properties characterized by the manufacturer, or determined experimentally (see [DOM10]). Figure 6.3 shows the evolution along the screw length of the average melt temperature, average degree of channel fill, cumulative mechanical power consumption, cumulative residence time, pressure and average shear rate. Table 6.2 contains the values of global process responses. Average strain is a measure of the extent of distributive mixing; Specific Mechanical Energy (SME) represents the mechanical energy consumption per unit weight of processed material; Viscous Dissipation quantifies the increase in melt temperature relative to the local set value. Not only the values of these parameters change significantly with screw geometry, but they are also conflicting. For example, the best distributive mixing is obtained for screw B (Table 6.2) at the cost of having to cope with higher viscous dissipation (which, if excessive, will cause premature material degradation). The performance measures presented in Table 6.2 will be adopted as objectives for the optimization runs discussed in subsequent sections.

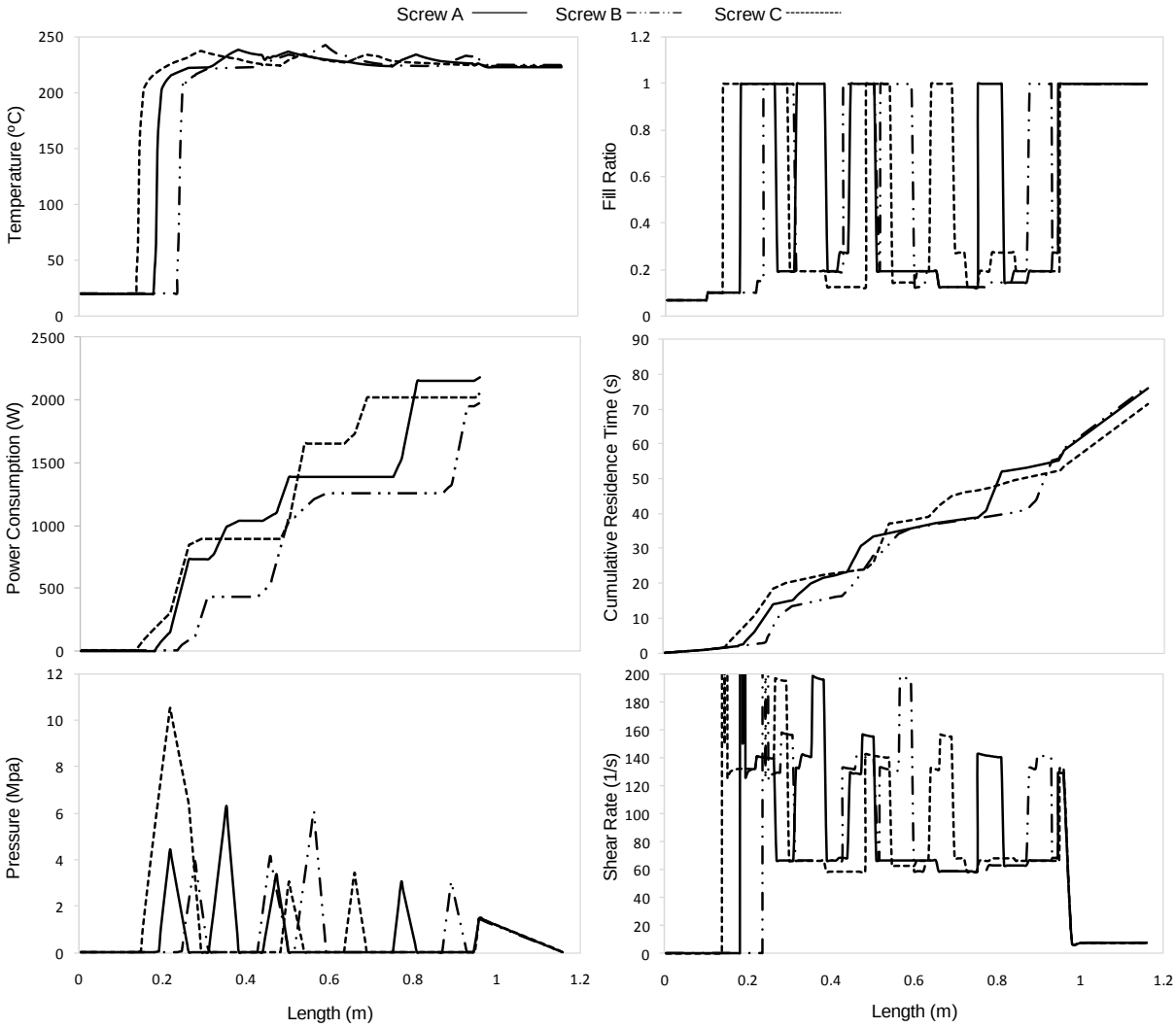

Figure 6.3. Evolution of various process parameters along the barrel when using the screw profiles of Figure 6.2 under identical operating conditions.

Table 6.2. Global process responses when using the screw profiles of Figure 6.2 under identical operating conditions.

	Average Strain	SME	Viscous Dissipation
Screw A	7449.9	0.979	1.198
Screw B	7682.6	0.889	1.217
Screw C	6859.4	0.925	1.198

6.2.3. The Problem to Solve

The examples presented above demonstrate the practical importance and difficulty in optimizing the process. In general, this should be done by using as parameters to optimize (i.e., the decision variables) the operating conditions [GAS02], the screw configuration [GAS05], the individual geometry of the different screw elements, or a combination of these variables, depending on the practical situation to tackle. The operating conditions and individual geometry of the elements are continuous variables, whilst the screw configuration variables are discrete (since they define the location of the elements through the order in which the screw elements are put on the screw shaft). Simultaneously, the best screw configuration can be determined for a fixed set of screw elements previously selected, or by using a pool of elements available for the machine under consideration [GAS05]. In the present work, the Twin-Screw Configuration Problem (TSCP) is deal with using a fixed number of pre-determined screw elements. Again, the TSCP consists in the definition of the best location along the screw axis of a number of distinct screw elements that optimizes the process performance. There are no restrictions in terms of the possible location of any screw element, except that the initial (upstream) elements must be of the conveying type (to ensure smooth material inlet) and at least one restrictive element (left handed or kneading block) must be present in order to melt the material. Thus, the TSCP can be seen as a sequencing problem where the resources (screw elements) must be ordered sequentially along the screw shaft.

For the purposes of this work, an instance of the problem consists of a pre-determined set of screw elements and their respective location along the screw shaft needs to be determined. For each instance, three different variants are defined by the particular combination of objectives to be considered. The objective values are computed by the modeling routine, i.e., and as seen above, for each screw configuration that is defined by the optimization algorithm, the modeling routine must be run. Each run of the modeling routine needs two to three CPU minutes (depending mainly on the number of restrictive screw elements) on an AMD opteron TM 2116 dual-core processor running at 2.4 GHz with 2MB L2-Cache. Hence, the total computation time for one run of an optimization algorithm with a maximum of 3000 evaluations of sequences by the modeling routine is more than 6000 CPU minutes, that is, more than 4 CPU days. Clearly, an exhaustive search for all possible sequences is infeasible for already a small number of screw elements and therefore heuristic algorithms are required to determine good screw configurations.

6.3. Multi-Objective Optimization Algorithms

6.3.1. Multi-Objective Evolutionary Algorithms

Most real-world optimization problems, including the TSCP, involve the simultaneous optimization of various criteria. The complexity of the task is increased by the existence of conflicting objectives [DEB01, COE07].

MOEAs are an efficient method to solve MOOPs: taking inspiration from the natural evolution process, they use a population of solutions that can approximate the Pareto front [DEB01, COE07].

However, the development of an Evolutionary Algorithm (EA) is strongly dependent on the characteristics of the problem to be solved [GOL89]. The most important characteristics to be considered in the present work are the discrete nature of the decision variables (location of the screw elements which can be represented as a permutation of the indices of the screw elements) and the presence of multiple conflicting objectives. Given the latter, the application of MOEAs seems to be straightforward. Different approaches for tackling MOOPs have been proposed in the literature [DEB01, COE07]. However, due to the first characteristic of the problem, none of the MOEA algorithms available in the literature can be used without considerable modifications concerning solution representation and the evolutionary operators. In fact, only the global concepts of these algorithms can be transferred to the TSCP.

Therefore, a modified version of RPSGA, a MOEA that was developed for continuous problems by Gaspar et al. was adopted here [GAS97, GAS00, GAS04]. In fact, in earlier research, RPSGA was found to be competitive with NSGA-II [DEB00], a standard MOEA proposed by Deb et al., which justifies this choice. RPSGA is based on the use of two populations (internal and external), that evolve simultaneously during the successive generations. RPSGA uses a clustering technique to rank the solutions present in the population. This technique replaces the crowding distance operator of NSGA-II and improves the quality of the solutions found when dealing with more than two objectives.

Due to the solution representation used for the TSCP (discrete variables that define a permutation of the screw element indices), the crossover and mutation operators in RPSGA were replaced by the inver-over operator [TAO98].

The main steps of the RPSGA algorithm are illustrated in Algorithm 4.1. As said above, RPSGA uses a clustering technique that reduces the number of solutions on the efficient frontier while

maintaining its characteristics intact; this enables the selection of the best solutions for reproduction and, simultaneously, it maintains the distribution of the solutions along the Pareto front [GAS00]. The clustering technique is applied by first dividing the set of solutions considered (either the internal or the external populations) by a pre-defined number of ranks (N_{ranks}). For the first rank (r equal to one), a clustering algorithm is applied until the number of solutions that rest is equal to (N/N_{ranks}), i.e., only this number of ranks exists. To these solutions, which represent the solutions that best represent the clusters, is attributed rank one (i.e., rank equal to r). For the second rank (r equal to two), the entire population is reduced to ($r*N/N_{ranks}$). To these individuals that do not have yet a rank (attributed in the first iteration), is attributed the rank two. This process is repeated until the last rank. To the non-dominated individuals is attributed the maximum rank value (i.e., N_{ranks}). A rank function is used to calculate a global evaluation function value and the individuals in the population are selected by a roulette wheel technique using this evaluation function value. Additionally, an archive of non-dominated solutions is kept, in order to prevent that good solutions are lost [GAS00]. For a more detailed description of RPSGA refer to [GAS97, GAS00, GAS04] and Subsection 4.4.2.

6.3.2. Multi-Objective Ant Colony Optimization

Ant Colony Optimization takes inspiration from the pheromone trail laying and following behavior of real ants and transfers some core behaviors to an algorithmic approach for tackling complex combinatorial problems [DOR99]. To apply ACO, pheromone trails are associated to solution components of the problem to be tackled.

Artificial ants then iteratively generate solutions to the problem under concern using a probabilistic solution construction mechanism and update the pheromone trails based on a positive feedback loop. From a high level perspective, the outline of an ACO algorithm is as follows (Algorithm 6.1).

The ACO algorithm starts with the initialization of the pheromone matrix. While the termination conditions are not met (i.e, at each iteration) the following tasks are carried out. First, a solution is constructed for each ant, based on probabilistic values that are a function of the pheromone strengths associated to solution components and possibly available heuristic information (step 2a). Then, each solution is evaluated (step 2b). Next, the pheromones are updated (step 2c) by first

reducing the pheromone trails by a fixed factor ρ (simulating pheromone evaporation) and then depositing some amount of pheromone on selected solution components. The role of pheromone evaporation is to allow forgetting of "poor" previous decisions and to help bias the search around the best solutions found. The role of the pheromone deposit is to favor the components of the best solutions in the generation of subsequent solutions.

If ACO is applied to MOOPs, non-dominated solutions are kept in an archive (step 2d) and all non-dominated solutions found during the search are returned. For this purpose, new non-dominated solutions are added to the archive and the solutions that become dominated are removed.

Algorithm 6.1. Generic ACO algorithm

(1) Initialize pheromone matrices to value τ_{ini} ;
(2) **while** termination conditions not met do
  (a) Construct solutions;
  (b) Evaluate solutions;
  (c) Update pheromones;
  (d) Update archive; % if ACO is applied to MOOPs in Pareto sense
end while

6.3.3. Algorithmic Components of MOACO algorithm for TSCP

When applying ACO algorithms to specific problems and to MOOPs, in particular, a number of implementation choices have to be taken [DOR99, DOR10, GAR07, ANG09, LOP09, LOP10b]. In the following, the details about the underlying ACO algorithm used, i.e., MAX-MIN Ant System, are presented, and then the algorithmic components considered to extend this underlying ACO algorithm to tackle the multi-objective version of the TSCP are discussed.

MAX-MIN Ant System for TSCP

As the ACO algorithm, on which the MOACO algorithm for the TSCP is based, the MAX-MIN Ant System (MMAS) is used [STU98b, STU00]. MMAS constructs solutions probabilistically by a standard construction mechanism that is common to most ACO algorithms. An ant starts with an empty solution and at each construction step it chooses probabilistically one solution component. In the classical action choice rule of ACO algorithms, which is also use in the present approach, an

ant m chooses solution component (i, j) with a probability given by:

$$p_{ij}^{m} = \frac{\tau_{ij}}{\sum_{l \in \aleph_i^l} [\tau_{il}]} \; if \quad j \in \aleph_i^m \tag{6.1}$$

where $\aleph_i^m$ is the feasible neighborhood of ant m (i.e., the screw elements that are still available in the TSCP case).

Once all ants have constructed a solution, the pheromone trails are updated. The update of the pheromone trails consists of the pheromone trail evaporation, which decreases the amount of pheromone by a fixed factor ρ, and the pheromone deposit. From a high level perspective, this pheromone update process can be described by:

$$\tau_{ij} = (1-\rho)\, \tau_{ij} + \Delta_{ij} \tag{6.2}$$

In the pheromone deposit, solution components occurring in one or several solutions generated by the ants increase their associated pheromone trail values. This in turn increases the probability that these solution components will be chosen subsequently by the ants in the following iterations. For the pheromone deposit, a first choice has to be made about which solutions are allowed to deposit pheromone on their solution components. One possibility is to consider only the best solution that is generated in the current iteration *(iteration-best strategy)*. Another possibility is to consider all the solutions generated since the start of the algorithm and to choose only the best of these *(best-so-far strategy)*. Clearly, the latter option results in a more directed search. The amount of pheromone deposited can then be defined, for example, in dependence of the quality of the solution generated.

A particularity of MMAS is that it restricts the level of the allowed pheromone trail values to a range $\tau_{\min} < \tau_{ij} < \tau_{\max}$. Using the pheromone trail limits reduces the possibility of search stagnation around the best solutions and favors the exploration of the search space [STU98b, STU00]. Following the general rationale of setting these values as explained in Stützle and Hoos [STU00], the values of $\tau_{\min}$ and $\tau_{\max}$ are set as follows:

$$\tau_{\max} = \frac{f_i}{\rho} \tag{6.3}$$

$$\tau_{\min} = \frac{\tau_{\max}}{2loc} \tag{6.4}$$

$$\tau_{ini} = \frac{\tau_{\max} + \tau_{\min}}{2} \tag{6.5}$$

where f_i is the best value of the objective function i, loc is the number of screw elements whose location must be defined and τ_{ini} is the initial pheromone value.

Finally, for applying MMAS to the TSCP, it is also needed to define the exact meaning of solution component (i, j) and, hence, how the pheromone trail information τ_{ij} is to be interpreted. In the case of the TSCP, a solution component may refer to an assignment of a screw element j to a specific position i on the screw axis *(position assignment).* In this case, τ_{ij} represents the desirability that screw element j is assigned to position i in the screw axis. Alternatively, a solution component (i, j) may refer to the successor relationship between screw elements on the screw axis *(relation assignment);* in this case, each entry τ_{ij} of the pheromone matrix represents the desirability that element j follows immediately after element i on the screw axis.

Considered Components for the MOACO algorithm

When applying ACO algorithms to MOOPs, a number of generic algorithmic components specific to the multi-objective aspects of the problem need to be defined. In the following the algorithmic components considered here are discussed.

Number of Pheromone Matrices and solution construction. When applying ACO to MOOPs, one may use either a single pheromone trail matrix to consider solutions for all objectives or multiple pheromone matrices [GAR07, LOP09]. (In the latter case, typically one pheromone trail matrix per objective is used.) The first option implies that the construction of the solutions follows the usual steps of the ACO algorithm, using the same probabilistic construction rules as in ACO algorithms for single-objective problems, which were already presented in the previous subsection.

When using several pheromone matrices, the information contained in the pheromone matrices needs to be aggregated for the probabilistic solution construction. Using a common generic form of this aggregation, an ant m then chooses a solution component (i, j) with a probability given by [IRE01, DOE04, LOP09, LOP10b]:

$$p_{ij}^{m} = \frac{\tau_{ij}^{\lambda_1} \otimes \tau_{ij}^{\lambda_2}}{\sum_{l \in \aleph_i^l} \left[\tau_{il}^{\lambda_1} \otimes \tau_{ij}^{\lambda_2} \right]} \quad if \quad j \in \aleph_i^m \tag{6.6}$$

where $\otimes$ is an aggregation operator, and λ_1 and λ_2 are the weights attributed to each objective, if two objectives are considered. The weights are normalized such that $\sum_i \lambda_i = 1$ (Equation 6.6 can easily be extended to more objectives.) Thus, the exploration of different regions of the Pareto front can be assured by the assignment of different weights to the ants. In the proposed algorithm is always assumed to have a maximally dispersed set of weight vectors.

Concerning the aggregation operator, two different choices have been taken in previous multi-objective ACO (MOACO) algorithms. One possibility is to use a weighted product aggregation *(product strategy),* i.e., in Equation 6.6 the operator $\otimes$ is actually $\times$. A second possibility is to use a weighted sum aggregation *(linear strategy),* i.e., in Equation 6.6 the operator $\otimes$ is replaced by $+$. Both choices were examined in the experimental analysis performed here.

Pheromone update. The pheromone update generally follows the steps explained already above for MMAS. The main issue to be considered in the multi-objective case is the choice of which solutions are used for the pheromone deposit.

Independently of whether one uses the iteration-best or the best-so-far strategy, the choice of the solutions for pheromone deposit depends on the number of pheromone matrices considered. If a *single* pheromone trail matrix for all objectives exists, the pheromone update is performed by using a maximum number of k non-dominated solution (or ants). Each of the k ants deposits an amount $1/k$ of pheromone.

The set of k non-dominated ants is selected through the use of a niching mechanism, in order that the solutions become evenly distributed along the Pareto front approximation [DEB89]. (To reflect the choice of the amount of pheromone to be deposited in this case, in the definition of $\tau_{\max}$ for the underlying MMAS algorithm, $\tau_{\max}$ is set equal to 2.0.) If *multiple* pheromone matrices are used, only a single pheromone matrix for each objective is used. In this case, the pheromone is deposited on the best solution (that is, on the iteration-best or the best-so-far solution, depending on which strategy is chosen) for each objective. The amount of pheromone deposited is then chosen to be proportional to the quality of the solutions.

Number of Colonies. Multiple colonies can be used in MOACO algorithms to better distribute the search effort along the Pareto front [IRE01]. Each colony is then specialized to search in a specific area of the Pareto front. For this purpose, the m ants are divided into p colonies. Each colony has its own pheromone information and the ants of each colony construct the solutions based only on their colony's pheromone information. The colonies then cooperate through the pheromone update mechanism. For example, the non-dominated solutions may be determined by considering the solutions generated by all colonies, imposing in this way a more selective determination of the non-dominated set of solutions.

In the pheromone update, cooperation can be achieved by the pheromone *update by region* scheme [IRE01]. In the bi-objective case considered here, the non-dominated solutions are first sorted along the front and then split into p subsets (p is the number of subsets, which is equivalent to the number of colonies). The best solutions of subset j are then used to update the pheromone information of colony j. In contrast, in the *update by origin* approach [IRE01], the ants of a specific colony are only able to update the pheromone trails of their own colony. Here, the update by region approach is adopted, since it is deemed to induce a more directed search.

The use of several colonies can be combined with MOACO approaches that use a single pheromone trail matrix or those that use several pheromone trail matrices. In this study, the second possibility is considered. (*A posteriori,* this choice is also justified by the fact that in the experimental analysis presented in Section 6.5 it turned out that the use of one pheromone trail matrix per objective resulted in better performance when compared to using only a single pheromone matrix.) When combining pheromone trail matrices by weights (λ), several possibilities of distributing the weights among colonies have been suggested [IRE01]. Here, 50% of the weights attributed to colony k are overlapped by the weights of colony $k+1$ and $k-1$.

6.4. Case Studies

As previously, the experiments are based on the characteristics of a Leistritz co-rotating twin-screw extruder, which is available at the University of Minho [GAS11]. A polypropylene homopolymer (ISPLEN PP 030G1E from Repsol) is being processed using a barrel and die temperature set to 220°C, the screws rotating at 150 rpm and a feed rate of 8 kg/hr. The goal is to define the best

location of the 16 screw elements identified in Table 4.1 such that performance is optimized. Four instances (TSCP1 to TSCP4) with a different number of restrictive screw elements (1 to 4, respectively) are considered and a total of 12 case studies are analyzed (Table 4.2) as described in subsection 4.2.2.

As stated before, the main goal of this work is to develop a high-performing MOACO algorithm for the multi-objective TSCP. For this purpose, first the several design alternatives of MOACO algorithms are tested, which have been discussed in the previous section and are summarized in Table 6.3.

Table 6.3. Components of the MOACO algorithm that have been studied.

ACO Component	Values tested
pheromone evaporation	0.1, 0.2, 0.3, 0.4 and 0.5
pheromone information	position *vs.* relation
pheromone update strategy	best-so-far *vs.* iteration
number of pheromone matrices	one *vs.* several
number of colonies	one *vs.* three
weight aggregation	linear *vs.* product

Then, the performance of the final MOACO algorithm to the previously developed RPSGA and a TPLS algorithm are compared. Since the computation time of the modeling routine for evaluating a solution is relatively high (between two and three minutes, depending on screw complexity), each algorithm run was limited to 3000 solution evaluations.

Given the stochastic nature of the optimizers, it is necessary to adopt statistical methods to compare the non-dominated solutions produced - the use of a single value would have severe limitations [ZIT03, KNO06]. For this reason, the performance of the algorithms was made using the Empirical Attainment Function (EAF) methodology [GRU01]. The EAFs are measured by running the multi-objective algorithms a number of times; here, ten independent runs are carried out with each algorithm. The performance of two algorithms can then be compared through plots of the differences of their EAFs [LOP06, LOP10a], that are indicated by points in the objective space, as described in subsection 2.2.3. Note that the points are only plotted in positions where the differences between the EAFs of the two algorithms change between different values. The values of

the differences are encoded using a grey scale; the darker the points, the larger the observed differences.

An example of such a plot is given in Figure 6.4. The two continuous border lines in the plot connect the best objective vectors that have been found in any of the runs of the two algorithms (grand best) and the points in the objective space that have been dominated by all runs of the two algorithms (grand worst); the discontinuous line in the middle gives the median of the EAF that was obtained by an algorithm (e.g., in Figure 6.4, left plot is given the median obtained by the best-so-far pheromone update strategy), i.e., the boundary of the objective space that was obtained in 50% of the algorithm runs. Figure 6.4 clearly indicates a better performance of the MOACO algorithm with best-so-far pheromone update strategy compared to the iteration best pheromone update strategy. In fact, all the differences between the EAFs are in favor of the best-so-far pheromone update strategy, some being rather large (between 0.8 and 1.0 as indicated by the black points).

6.5. Results and Discussion

First, the influence of the MOACO design choices and parameter settings on performance, using the three case studies associated to instance TSCP4, is analyzed. Then, the best MOACO results to RPSGA and TPLS on all twelve case studies (using all instances TSCP1 to TSCP4) are compared. In the experiments presented here, the colonies used a cross-total of 50 ants (except in the case were the influence of the number of colonies is studied) and 60 iterations were performed, resulting in 3000 solution evaluations with the modeling routine.

6.5.1. Influence of MOACO components

In this section, a summary of the main results obtained when comparing different design choices and parameter settings of the MOACO algorithm is presented. Since the presentation of all EAF differences plots would be too extensive, only a few illustrative results using instance TSCP4 and case studies 1 and 2 (see also Table 4.2) are reported here; the complete set of results is available in Appendix B.

Pheromone information

The comparison between the relation and the position-based pheromone information (Figure B.1 in Appendix B) using the plots of the EAF differences demonstrate that almost all differences are in favor of the relation-based definition of the pheromone information. These are expected results, since in the polymer extrusion process studied here there is a strong interaction between contiguous elements, mainly when restrictive elements are considered.

The performance of the position-based pheromone information could also not be further improved by the application of additional techniques. For example, the application of the pheromone summation rule is considered [MER00], but no performance improvement was obtained (plots shown in Appendix B, Figure B.2). As a further refinement, it is considered the choice of an assignment order, which gives preference to first assign screw elements (*e.g.*, kneading disks and left handed elements), which are supposed to have a more important impact on the global process characteristics than others. However, in the tests on the assignment order of the screw elements (sequential vs. importance-based), a consistent improvement by using this "importance-based" assignment order could not be obtained (see Figures B.3 and B.4 - Appendix B). Hence, neither the summation rule nor the importance-based assignment order was included into our final MOACO algorithm. For the following results, the use of the relation-based definition of the pheromone information is always used.

Pheromone update strategy

Figure 6.4 presents the results obtained for the two different pheromone update strategies tested. The best-so-far update strategy (best among the solutions existing in the archive) produces much better results than the use of the iteration-best update strategy and, hence, it will be adopted in other experiments. See Figure B.5 in Appendix B for complete results. Additionally, the influence of the pheromone trail evaporation factor ρ (Figures B.6 to B.15 - Appendix B) is examined. The best results were obtained for values of $\rho = 0.2$ and $\rho = 0.3$.

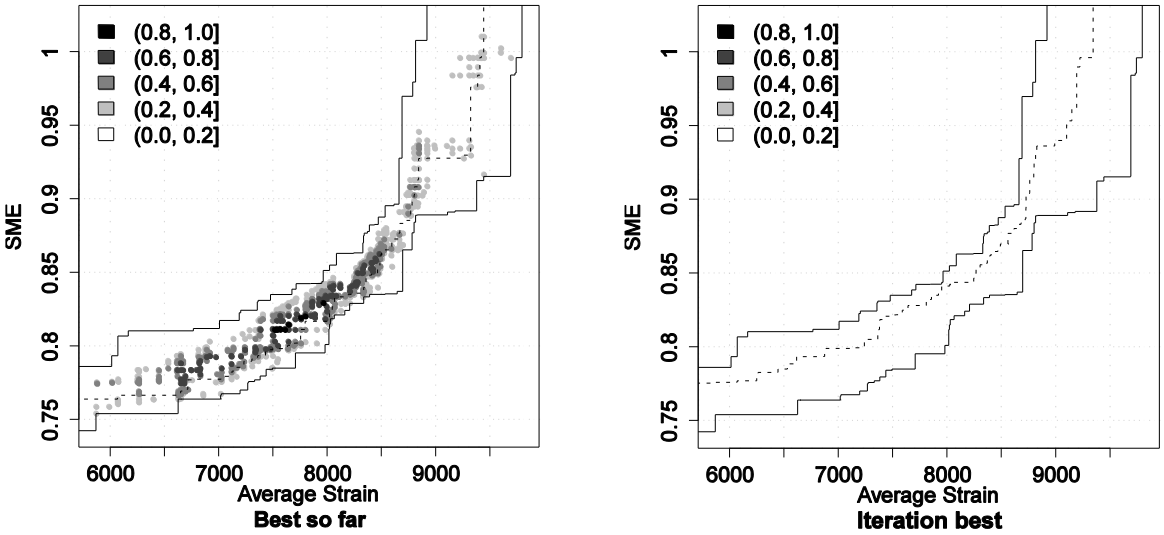

Figure 6.4. Influence of the pheromone update strategy (best-so-far *versus* iteration-best).

Number of pheromone matrices

As anticipated, the utilization of several pheromone matrices, using one matrix for each objective together with weighted aggregation of the pheromone trail values, produced significantly better results than the adoption of a single pheromone trail matrix (see Figure B.16 - Appendix B).

Number of colonies

Two different conditions were analyzed: the application of a single colony with 50 ants during 60 iterations and the application of three colonies with 20 ants each during 50 iterations. In both cases, each colony uses two pheromone matrices. (Note that for each of the three colonies 20 ants and 50 iterations were used to limit the computational effort to 3000 solution evaluations by the modeling routine.) Better results were attained in the second case, as depicted in Figure 6.5 and Figure B.17 (Appendix B).

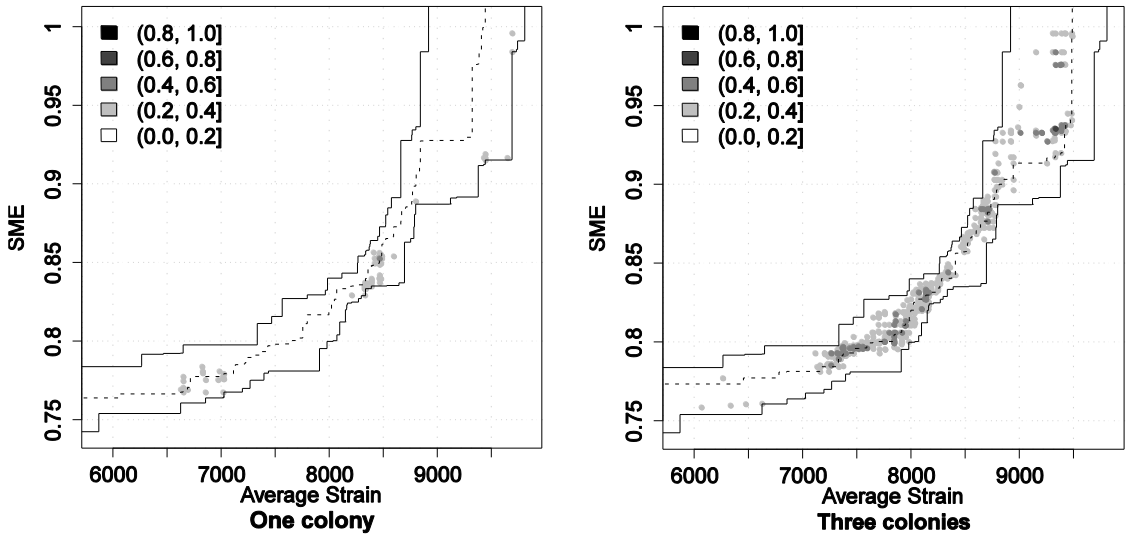

Figure 6.5. Influence of the number of colonies (one colony *versus* three colonies).

Weight aggregation

Finally, as seen in Figure 6.6, it can be concluded that the product strategy for the pheromone aggregation produces substantially better results than the weighted sum aggregation by the linear method. See Appendix B, Figure B.18 for complete results.

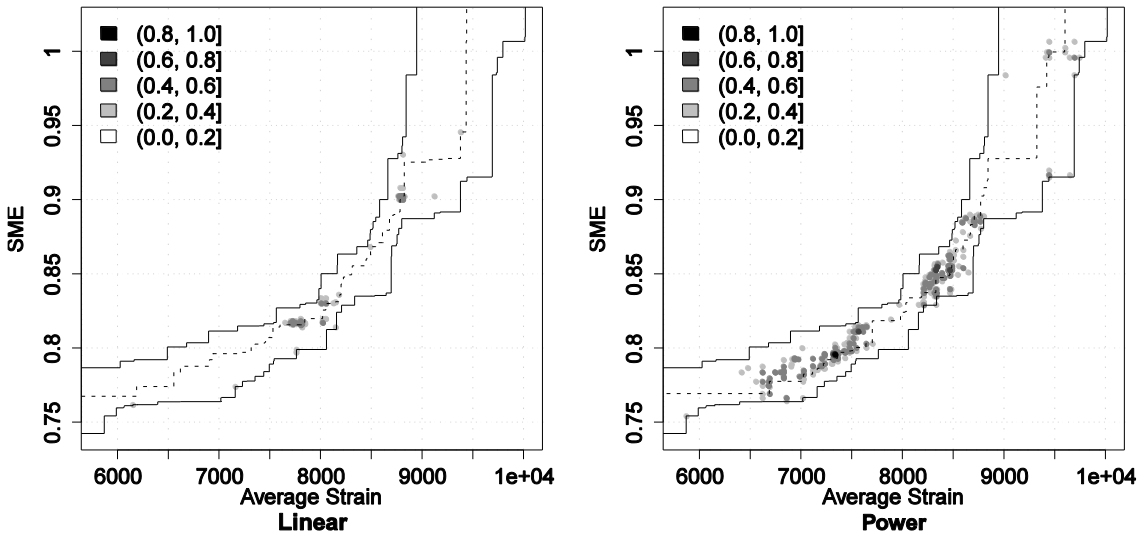

Figure 6.6. Influence of the weight aggregation method (linear *versus* product).

A summary of the above results is presented in Table 6.4. The second column represents the best values obtained when the parameters are studied individually. These best values are used for the final MOACO algorithm, the performance which is compared with the RPSGA in the next section.

Table 6.4. Best parameters values suggested from experimental analysis for the MOACO algorithm.

MOACO Component	Values tested
pheromone evaporation rate	0.2
pheromone information	relation
pheromone update strategy	best-so-far
number of pheromone matrices	several
number of colonies	three
weight aggregation	product

6.5.2. Comparison with RPSGA and TPLS

Here, the performance of the MOACO algorithm (using the best combination of parameters, as presented in Table 6.4) is compared with RPSGA and TPLS. Each algorithm is run for a maximum of 3000 solution evaluations on each of the 12 case studies of the bi-objective TSCP.

The differences among the empirical EAFs can be observed in Figure 6.7 for the case of the TSCP4 instance (the corresponding Figures for instances TSCP1 to TSCP3 can be consulted in Appendix B; Figures B.19, B.20 and B.21, respectively), where each figure relates to the results comparing the three case studies for a same number of restrictive elements to be sequenced (see also Table 4.1). The differences between the algorithms are higher for the case studies on TSCP3 and TSCP4 with three and four restrictive elements, respectively. In fact, in this case the advantages are almost exclusively in favor of the MOACO algorithm. On the case studies related to TSCP1 and TSCP2, still most of the differences are in favor of the MOACO algorithm developed, although they are typically limited to smaller areas of the objective space. This outcome seems to indicate that the advantage of MOACO over RPSGA increases with a growing number of restrictive screw elements.

As a further analysis, the hypervolume indicator was also computed [ZIT03]. For minimization problems, the hypervolume in two dimensions measures the surface that is dominated by the non-dominated solutions of a Pareto-front approximation and bounded by a point that is larger in every objective than any of the solutions in the Pareto-front approximation. The case of maximization problems can be described analogously. Given the very large differences of the ranges between the three objectives, all results are first normalized into the interval [1, 2]. This is done by first converting the maximization objective into a minimization one and then considering for

each objective and each instance the smallest and largest objective values found by any of the two algorithms. The smallest value is then mapped to 1.0 and the largest to 2.0. Simultaneously, the dominated points are removed. The hypervolume is then computed taking the worst point, (2.0, 2.0) as a reference. Using the resulting hypervolume values, a Wilcoxon signed rank test was further conducted to examine the statistical significance of the observed differences. The results are summarized in Table 6.5, were the average hypervolume values for each of RPSGA and MOACO measured across the 10 independent runs of each algorithm and the *p*-value of the statistical test are presented. The larger of the two hypervolume values is indicated in italics font; if the differences are significant at the usual 0.05 significance level, this is noted in boldface. As can be seen from these results, in all case studies, the MOACO algorithm obtains a higher average hypervolume value than RPSGA, indicating its superiority over RPSGA. In fact, a binomial test, which has as null hypothesis that RPSGA and MOACO have the same probability of reaching a smaller average hypervolume than the other, is rejected in favor of MOACO (*p*-value = 0.00024). Hence, it is possible to conclude that MOACO reaches for the case studies considered better performance than RPSGA. If given in turn attention for each individual comparison on each case study, it is possible to observe that on four of the twelve case studies, the difference in hypervolume is statistically significant in favor of MOACO (and none is in favor of RPSGA). This is noteworthy, since by having only 10 independent runs of each algorithm, the power of the test is not very high.

Finally, the performance of the MOACO algorithm was compared with the earlier results obtained with the TPLS algorithm [TEI11]. (The detailed results with the plots of EAF differences can be found in Appendix B in Figures B.22 - B.25). The same conclusions can be drawn, i.e., MOACO is superior to its competitor in most case studies. (In fact, only for one case study - instance TSCP4, case study 1, involving average strain and SME objectives, TPLS has an advantage over the MOACO algorithm). Hence, the proposed MOACO algorithm is the current method of choice to tackle the TSCP.

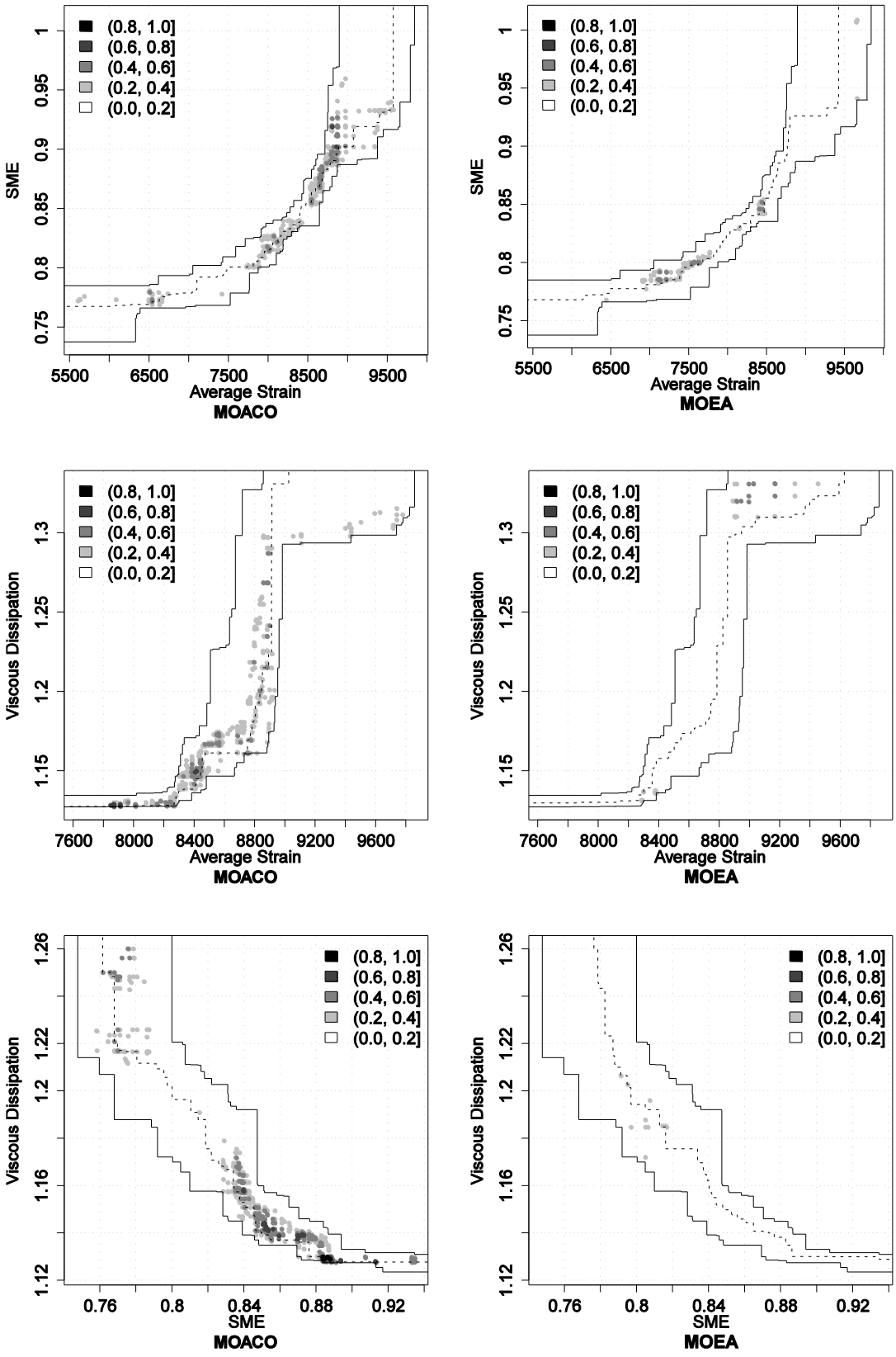

Figure 6.7. Comparison between MOACO and RPSGA for TSCP4 instance.

Table 6.5. Comparison of normalized hypervolume values between RPSGA and MOACO. An entry marked in italic face indicates a higher hypervolume; an entry marked in bold face indicates a statistically significant difference according to the Wilcoxon signed rank test in favour of one algorithm. In the last column are indicated the observed *p*-values for the Wilcoxon test.

Instance	Objectives	MOACO	RPSGA	*p*-value
TSCP1	Avg Strain, SME	*0.9502*	0.9421	0.1403
	AvG. Strain, Vis. Dissipation	*0.9039*	0.8777	0.9705
	SME, Vis. Dissipation	***0.6563***	0.6413	1.083e-05
TSCP2	Avg Strain, SME	***0.8432***	0.8147	0.0311
	AvG. Strain, Vis. Dissipation	*0.7570*	0.7431	0.2526
	SME, Vis. Dissipation	*0.7522*	0.7418	0.6772
TSCP3	Avg Strain, SME	***0.8554***	0.8070	0.0041
	AvG. Strain, Vis. Dissipation	***0.8156***	0.7368	0.0039
	SME, Vis. Dissipation	*0.6975*	0.6805	0.1655
TSCP4	Avg Strain, SME	*0.7618*	0.7439	0.3073
	AvG. Strain, Vis. Dissipation	*0.5361*	0.5042	0.2475
	SME, Vis. Dissipation	*0.6563*	0.6154	0.0630

6.6. Conclusions

An effective MOACO algorithm for the bi-objective TSCP was presented. First, a number of different design choices and parameter settings of the MOACO algorithm were examined. Based on the knowledge gained with that study, a MOACO algorithm for the bi-objective TSCP is proposed and its performance is evaluated on a total of 12 case studies. The analysis of the experimental results via the use of state-of-the-art methods for evaluating the performance of multi-objective optimizers demonstrated that the proposed MOACO algorithm is the method of choice for tackling the TSCP. Effective algorithmic tools to define appropriate screw configurations are indeed required by the industry.

There are a number of possible directions where to extend this research. The first is to consider combinations of MOACO algorithms with TPLS, or other local search methods, to further improve the results. Given the high computation times required by the modeling routine to evaluate the screw configurations, another promising direction would be to reduce the number of calls to the

modeling routine by using statistical modeling techniques for predicting the objective function values. Finally, extending the work to embrace the situation where the user must choose the appropriate screw elements from a larger set of available elements would be very interesting from a practical standpoint. Actually, in this case two interrelated problems arise. The first is to select the screw elements from a larger set of available ones, while the second is to define the sequence of elements on the screw shaft.

Chapter 7

Hybrid Algorithms

In this chapter, different hybrid algorithms considering the algorithms studied previously (TLS, PLS, RPSGA and MOACO) were developed to tackle bi-objective versions of Twin Screw Configuration Problem (TSCP). Two distinct approaches were considered. One is to use the PLS algorithm as a form of post-processing the solutions generated by the other three algorithms. The other one is to run, in a first step, the TPLS algorithm to provide a good initial set of solutions. These solutions will be the initial population of the main algorithm.

The several hybrid algorithms and the underlying pure search strategies were compared making use of the Empirical Attainment Functions (EAF). An improvement of the performance of MOACO and RPSGA algorithms was obtained when linked with local search procedures (TPLS and PLS). Finally, a statistical analysis using Friedman tests was applied to compare the performance of all the algorithms studied in this work. MOACO+PLS reveals to be the best algorithm to tackle TSCP.

7.1. Introduction

In previous chapters, a series of efforts have been made to develop an efficient multi-objective algorithm to tackle the Twin Screw Configuration Problem (TSCP).

First, several tests were performed in order to obtain an effective single objective iterative improvement algorithm: study of neighborhood structure, neighborhood search strategies and neighborhood restrictions. Then, the single objective SLS algorithm was extended to several bi-objective case studies of the TSCP by the application of Two Phase Local Search algorithm [TEI11] and Pareto Local Search algorithm [TEI10b]. TPLS is based on a search model using a scalarized acceptance criterion (SAC) whereas PLS is based on a search models using a component-wise acceptance criterion (CWAC) [PAQ07]. In both cases, good results were obtained when compared with a Multi-Objective Evolutionary Algorithm (Reduced Pareto Set Genetic Algorithm – RPSGA) previously proposed. In another line of research, Ant Colony Optimization (ACO) algorithms were adapted to tackle the TSCP.

For that, several ACO algorithm parameters were studied and tested in order to obtain an efficient ACO algorithm and the final ACO configuration was compared with RPSGA and TPLS algorithms. Very promising results, often superior to RPSGA, were obtained [TEI12a].

All the algorithms developed so far for the TSCP problem, rely on a single type of search method such as local search, evolutionary algorithms or ant colony optimization.

However, given the high computation time required by the modeling software developed to evaluate one single solution [TEI07, TEI12b], an as efficient as possible optimization algorithm needs to be developed in order to obtain good screw configurations in reasonable computational time. To accomplish this task, the combination of various search methods into one algorithm is considered in this chapter.

Hybrid algorithms are characterized by combining components of different algorithmic ideas trying to combine the advantages of several methods and this way to either speed up the search process or to improve the quality of the solutions that are obtained after a same computation time. An overview of different possible ways of combining algorithms can be found in [TAL02, EHR08, BLU08, BLU10]. In fact, in many cases, it was shown that by the combination of several search paradigms into a single algorithm, improved performance could be obtained. The probably most common approach of generating hybrid methods is to complement population-based metaheuristics with local search procedures [MUH88, ULD91, MOS99, DOR04] which tries to

overcome the disadvantage of slow convergence and missing local fine-tuning abilities, characteristic of many population-based methods.

In this chapter, we examine whether the performance of the heuristic algorithms we have developed previously can be further improved by considering their hybridization.

Various possibilities are examined. As a first possible hybrid, the post-processing of solutions generated by RPSGA (MOEA), TPLS and multi-objective ACO (MOACO) algorithms by applying PLS are considered. In particular, the set of non-dominated solutions generated by RPSGA, TPLS and MOACO serves as the initial solution set of PLS. In particular, the combination of TPLS and PLS has already previously been studied for various combinatorial optimization problems, in part leading to substantial improvements over the individual search methods [PAQ07, DUB13]. As a second possibility, the usage of TPLS to seed the population-based methods RPSGA and MOACO with good quality initial solutions is considered, trying to speed-up their convergence to high-quality solutions. Overall, all the hybrid algorithms can be characterized as hybrids that work in two distinct phases. In a first phase, some underlying search method is applied, which generates some initial set S of non-dominated solutions. This set S is then improved in a second phase by one of the other search methods.

The various hybrid methods and the underlying pure search strategies are compared using a benchmark set of twelve case studies, as done in previously works (see Section 4.2.2 for details). The experiments are set-up in such a way that all methods are given a same number of solution evaluations, thus making the comparison fair in this sense. The experimental results show that generally the hybrid algorithms perform better than the main underlying single search strategies used in the first phase which they are. Take this into account and the good results obtained in previously studies, the aim of this work is to develop a high performance hybrid algorithm to deal with the TSCP using local search procedures (TPLS and PLS) into population based metaheuristics, as MOACO and MOEA algorithms previously developed. A combination of TPLS and PLS was also studied.

This chapter is organized as follows. The metaheuristics studied are explained in Section 7.2 and the results are discussed in Section 7.3. Finally, the main conclusions are presented in Section 7.4.

7.2. Hybrid Algorithms

In earlier research, TPLS, PLS, MOEA (RPSGA) and MOACO algorithms were studied mainly in isolation [TEI10b, TEI11, TEI12a]. As a natural extension, the possibility of using hybrid algorithms that combine two of these algorithms is considered. For the hybridization, the composition of a new hybrid algorithm from a concatenation of two of the search methods is considered. In principle, two main approaches are followed and explained in more detail next. The first is to post-process the solutions generated by TPLS, MOACO, or MOEA by using a PLS algorithm and the second is to seed the initial set of candidate solutions by the non-dominated solutions generated in short runs of TPLS. These candidate solution build then a seed population for MOACO and MOEA (and PLS), respectively.

7.2.1. Improving solutions with PLS

One natural choice is to use the PLS algorithm as a form of post-processing the solutions generated by the other three algorithms. If the initial set of high-quality solutions is generated by TPLS, a hybrid TPLS+PLS algorithm [DUB13] is obtained. If MOEA or MOACO are used to generate the initial set of candidate solutions for PLS, the resulting algorithms are denoted as MOEA+PLS and MOACO+PLS, respectively. Generally, the goal of this type of hybrids is to improve the quality of the solutions by refining the set of solutions returned by either of the three methods. TPLS, MOACO and MOEA are in this case stopped early to allow comparing the solution quality reached by the hybrid algorithms to the pure strategies using a same computational effort. Note that, for TPLS, MOEA, and MOACO an archive with all non-dominated solutions found during the search process is saved. Once these methods reach a certain number of evaluations, pre-defined by the user, PLS is started from the non-dominated solutions found by the respective first metaheuristic used in the hybrid.

7.2.2. Seeding population-based Algorithms with TPLS

The quality of the initial population can have a significant influence on the search performance of population-based algorithms and therefore a common goal in memetic algorithms [MOS99] is to generate high quality initial population. The usage of good quality initial populations usually has an advantage over random initial populations at least for what concerns the quality of the solutions

generated in the early iterations of MOEAs and MOACO algorithms. Here, this goal is followed by using TPLS to provide a good initial set of solutions (chapter 4). In a first phase, TPLS is run with four pre-defined weights $\lambda \in \{0, 1/3, 2/3, 1\}$, which results in weight vectors (0; 1); (1/3; 2/3); (2/3; 1/3); (1; 0). TPLS stops when the best solution for each weight is found.

When used with MOACO, resulting in the hybrid TPLS+MOACO algorithm, for each weight vector the respective pheromone matrices are updated with an amount of pheromone corresponding to five times the solution's final quality. Thus, when the MOACO phase of the hybrid algorithm starts, the ants' solution construction already takes into account information on the solutions found by the TPLS algorithm. In other words, the TPLS solutions are used to bias the solution construction during the MOACO algorithm. When used with MOEA, resulting in the hybrid TPLS+MOEA algorithm, the initial population of the MOEA algorithm will have included the non-dominated solutions generated during the execution of TPLS; if the population cannot be filled with the non-dominated solutions generated by TPLS, it will be completed by solutions that are generated uniformly at random.

Finally, it is worthwhile to mention that there are actually two interpretations for the TPLS+PLS hybrid. The first is that PLS is used to post-process the non-dominated solutions generated by TPLS, while the same hybrid can also be seen as using TPLS to seed the PLS algorithm with high-quality initial solutions instead of using a single random solution as seed for PLS.

7.3. Results and Discussion

In this section, the performance of the various heuristic algorithms considered in this study is compared.

Considering the fact that the algorithms applied are randomized, each of the algorithms was run 10 independent times on each of the 12 case studies (Table 4.1 and 4.2). As a first step of this analysis, the usage of attainment functions was considered, which were proposed by Fonseca and Fleming [FON96]. To allow the visualization of the regions where the differences between two optimizers are significant, the points in the objective space where the differences between the empirical attainment functions [FON01] of two algorithms are significant will be plotted [LOP10a]. In order to make a fair comparison with the algorithms studied previously, the estimation of the EAF uses the 10 runs of each algorithm with a different random number seed. Each algorithm was run

for 3000 evaluations of the modeling routine on each of the 12 case studies as done in previous studies. Given the large number of plots, only examples obtained from instance TSCP4 are presented in this chapter. The EAF plots on all instances can be consulted in Appendix C.

In a second step, the performance of the heuristic algorithms is summarized using performance indicators for multi-objective optimization. In particular, we use the hypervolume indicator that was suggested by Zitzler et al. [ZIT99] as one of the very few unary performance indicators that complies best with the principle of Pareto optimality. The normalization of the results obtained by each algorithm was done as described in Section 6.5.2. Where statistical tests are used, a threshold for the error of first type of α= 0.05 is assumed as default.

7.3.1. Improving solutions by PLS

As a first step in the experimental analysis, the differences in the empirical attainment functions between the single, non-hybrid search methods and the hybrids that use PLS for post-processing the solutions generated by TPLS, MOACO and MOEA are examined. Recall that all comparison are based on the same computational effort, that is, each of the algorithms compared is allowed to generate the same maximum number of screw configurations during a run.

Each of the algorithms is first considered in turn. Figure 7.1 shows the EAF differences between MOACO+PLS and MOACO for the three case studies of TSCP4. As can be seen, the advantage of MOACO+PLS is rather strong. In fact, the advantage of MOACO+PLS over MOACO becomes stronger as the number of restrictive screw elements increases from one in TSCP1 to four in TSCP4 (see also Figures C.1–C.3 in Appendix C).

The comparison of the performance of MOEA+PLS with the original MOEA for instance TSCP4 is presented in Figure 7.2. Similar to the MOACO+PLS hybrid algorithm, the combination of PLS with MOEA increases the performance of MOEA algorithm with a substantial increase of differences as the number of restrictive screw elements increases (see Appendix C, Figures C.4–C.6).

The same trend as for MOACO and MOEA, although less strong, is also observable for the TPLS+PLS hybrid when compared to executing TPLS alone. The EAF differences for the three case studies of TSCP4 can be seen in Figure 7.3. Supplementary results are presented in Appendix C, Figures C.7-C.9.

Hence, an examination of the EAF difference plots indicates that the usage of PLS as a post-processing of the solutions generated by the other search methods, that is, MOACO, MOEA, and TPLS, generally improves performance.

7.3.2. Seeding algorithms by TPLS

Next, the hybrid algorithms that result by seeding the initial set of solutions of MOACO, MOEA, (and PLS) by using the non-dominated solutions returned by TPLS is considered. When comparing TPLS+MOACO and TPLS+MOEA to MOACO and MOEA, respectively, the conclusions are more mixed. In fact, for few case studies advantages of the hybrid algorithms over respectively MOACO and MOEA are observed, though for other case studies the EAF differences rather indicate a minor advantage for the non-hybrid search methods. The computational results for TSCP4 can be observed in Figures 7.4 and 7.5, respectively. The remain results can be consulted in Appendix C, Figures C.10–C.15. The overall, general trend is that for a larger number of restrictive screw elements (in particular, three or four), the hybrid algorithms appear to be better while for few restrictive screw elements (in particular, one or two), the non-hybrid algorithms are competitive or sometimes slightly better.

As mention in Section 7.2.2., TPLS+PLS can be seen as seeding PLS by TPLS or as post-processing TPLS by a PLS algorithm. If this second perspective is considered, the performance of TPLS+PLS should be compared to that of PLS alone. In the latter case, PLS is seeded by one randomly generated configuration. In this case, there seems to be a slight edge of the hybrid algorithm over PLS, though the differences between the two do not appear to be very strong (see Figure 7.6 and Appendix C, Figures C.16–C.18).

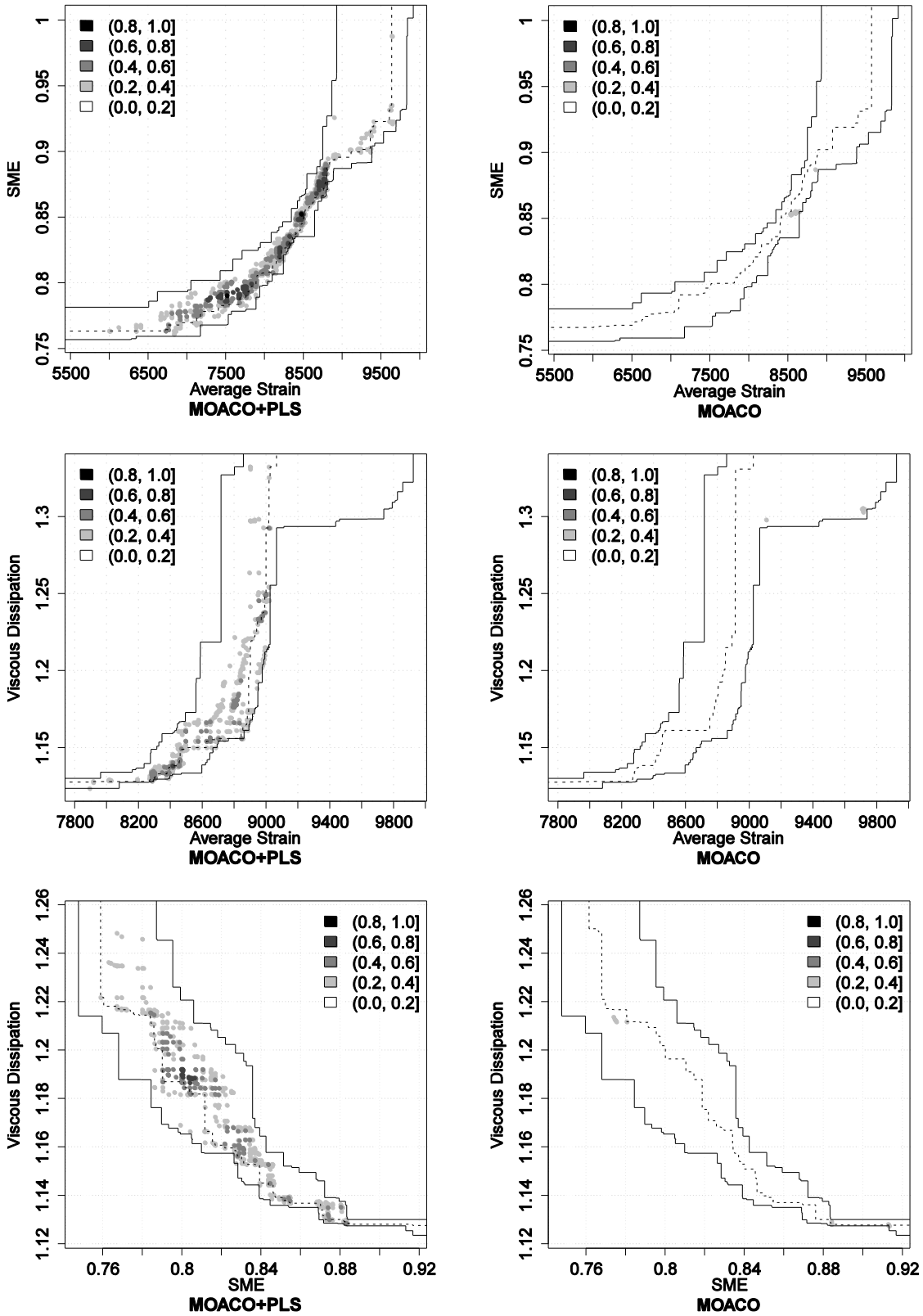

Figure 7.1. Results for instance TSCP4. Differences in terms of empirical attainment functions between the MOACO+PLS algorithm and MOACO after 3000 evaluations of the simulation program. Advantages in favor of MOACO+PLS are indicated on the left side; those in favor of MOACO on the right side.

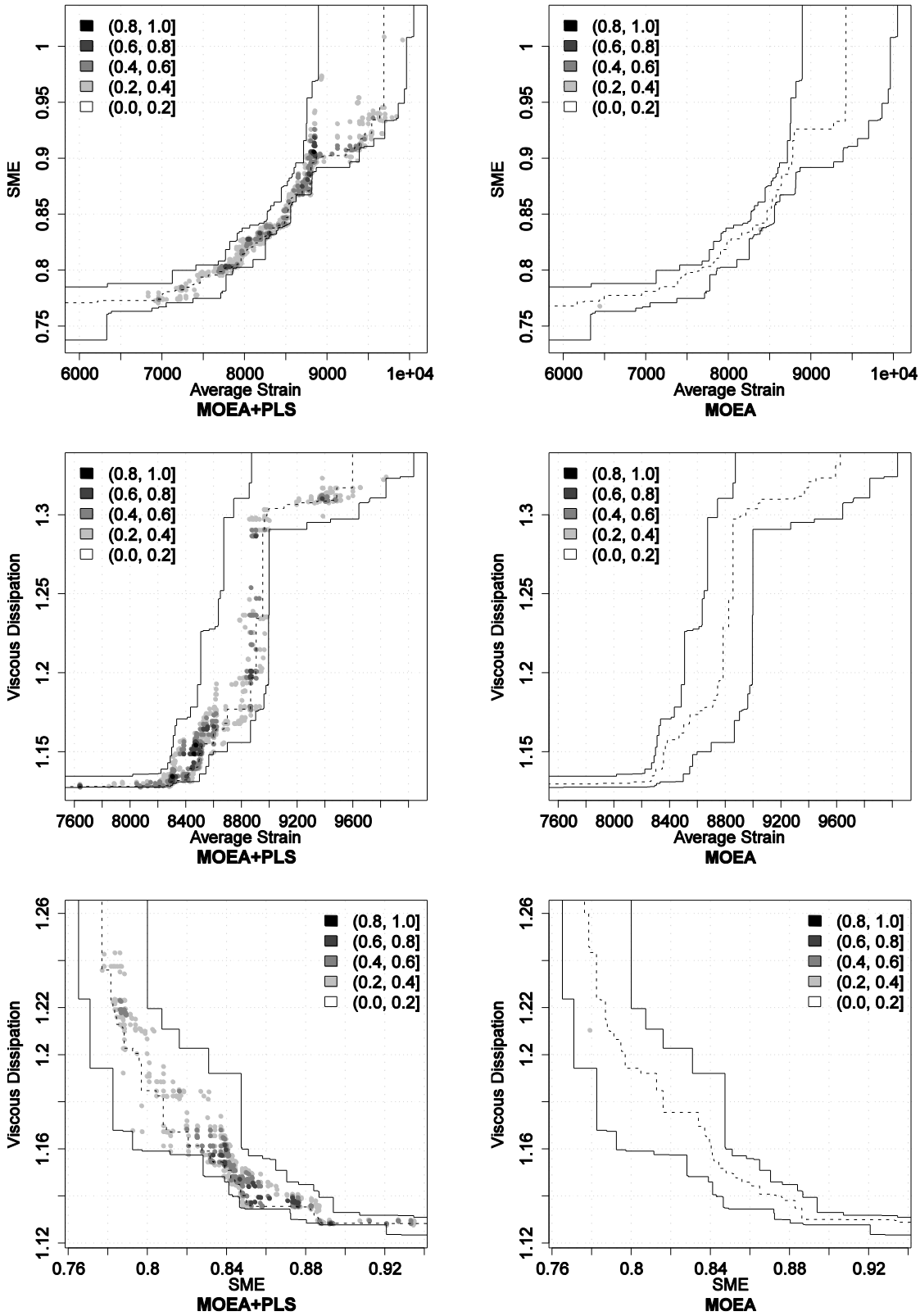

Figure 7.2. Results for instance TSCP4. Differences in terms of empirical attainment functions between the MOEA+PLS algorithm and MOEA after 3000 evaluations of the simulation program. Advantages in favor of MOACO+PLS are indicated on the left side; those in favor of MOACO on the right side.

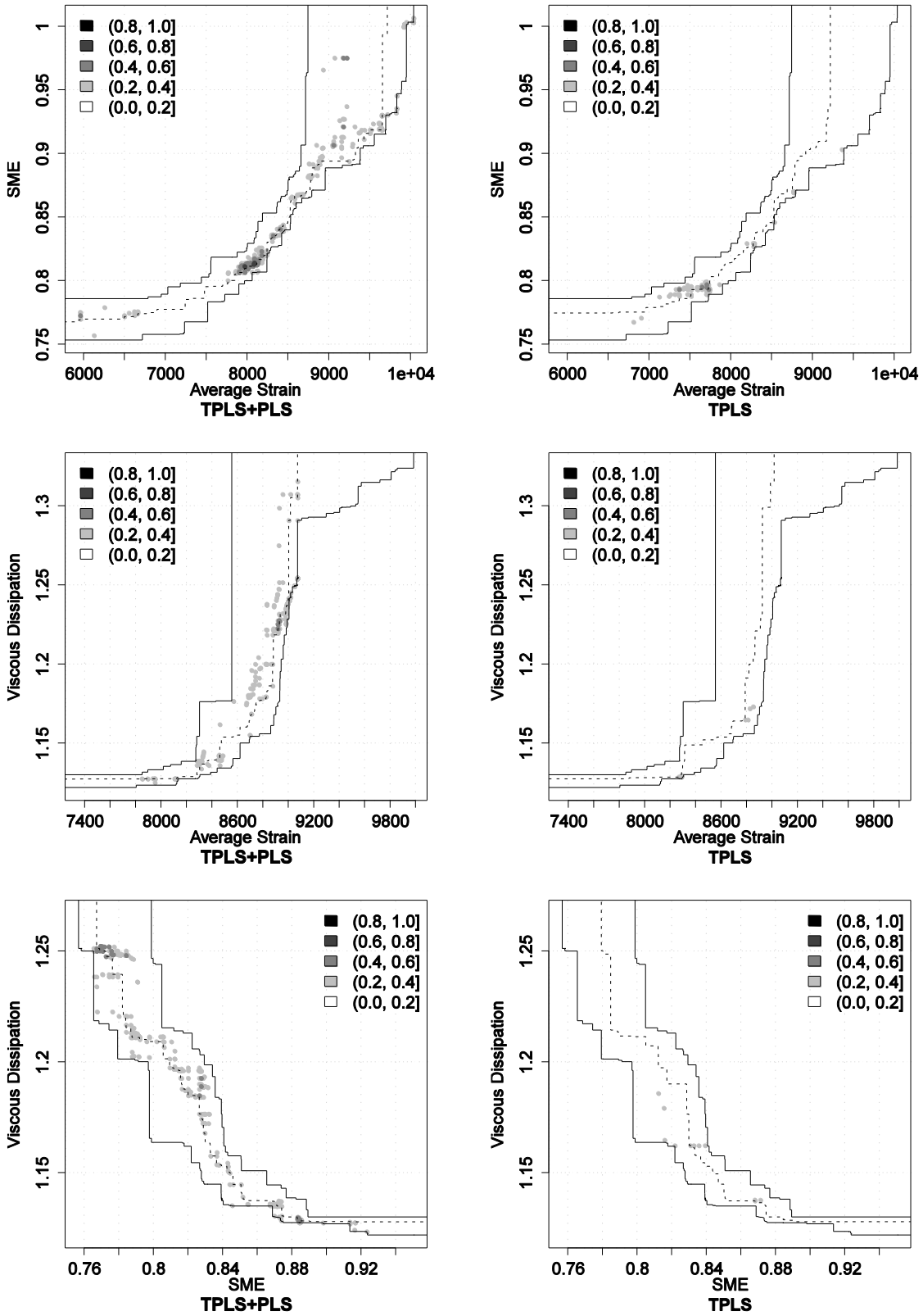

Figure 7.3. Results for instance TSCP4. Differences in terms of empirical attainment functions between the TPLS+PLS algorithm and TPLS after 3000 evaluations of the simulation program. Advantages in favor of TPLS+PLS are indicated on the left side; those in favor of TPLS on the right side.

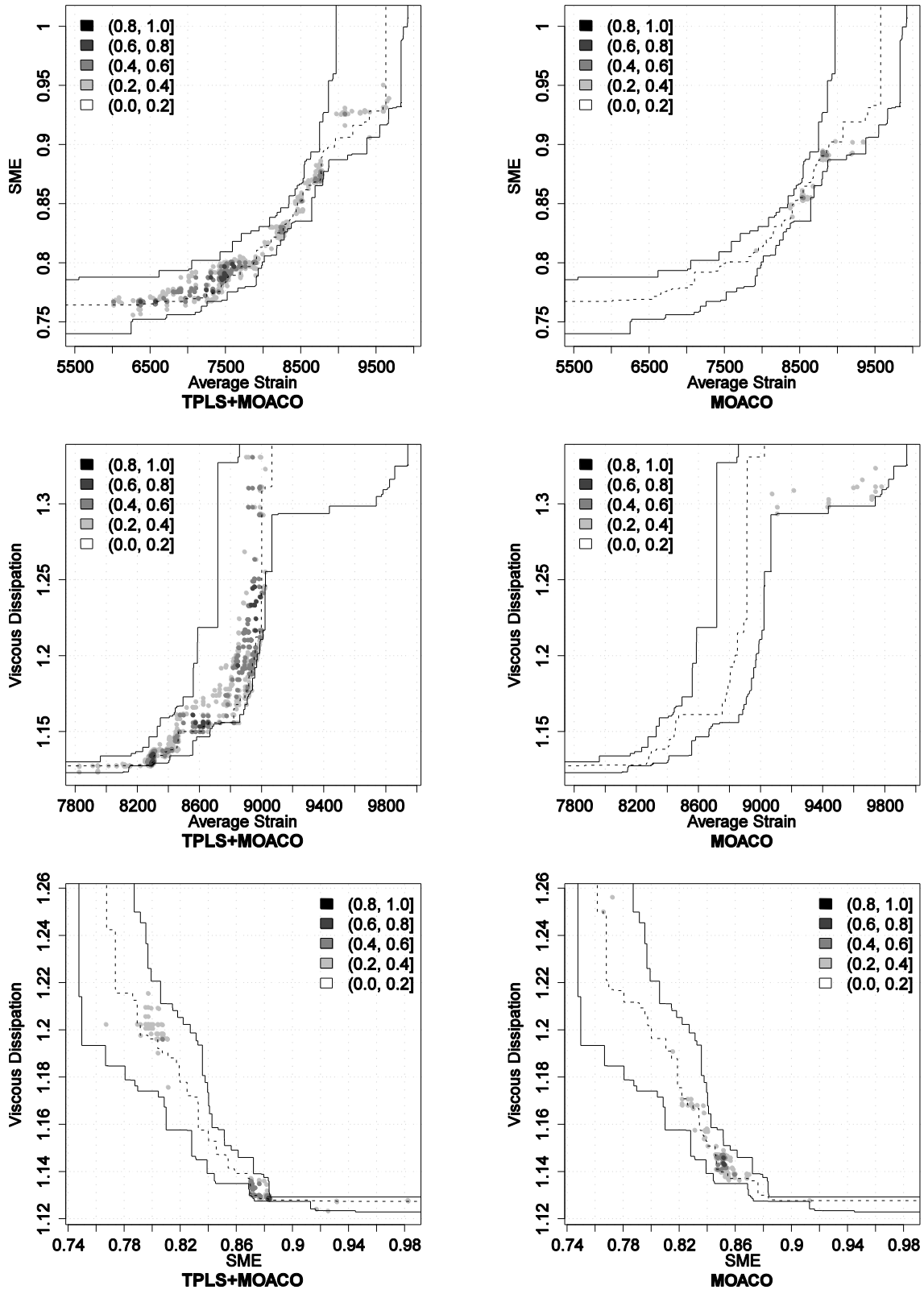

Figure 7.4. Results for instance TSCP4. Differences in terms of empirical attainment functions between the TPLS+MOACO algorithm and MOACO after 3000 evaluations of the simulation program. Advantages in favor of TPLS+MOACO are indicated on the left side; those in favor of MOACO on the right side.

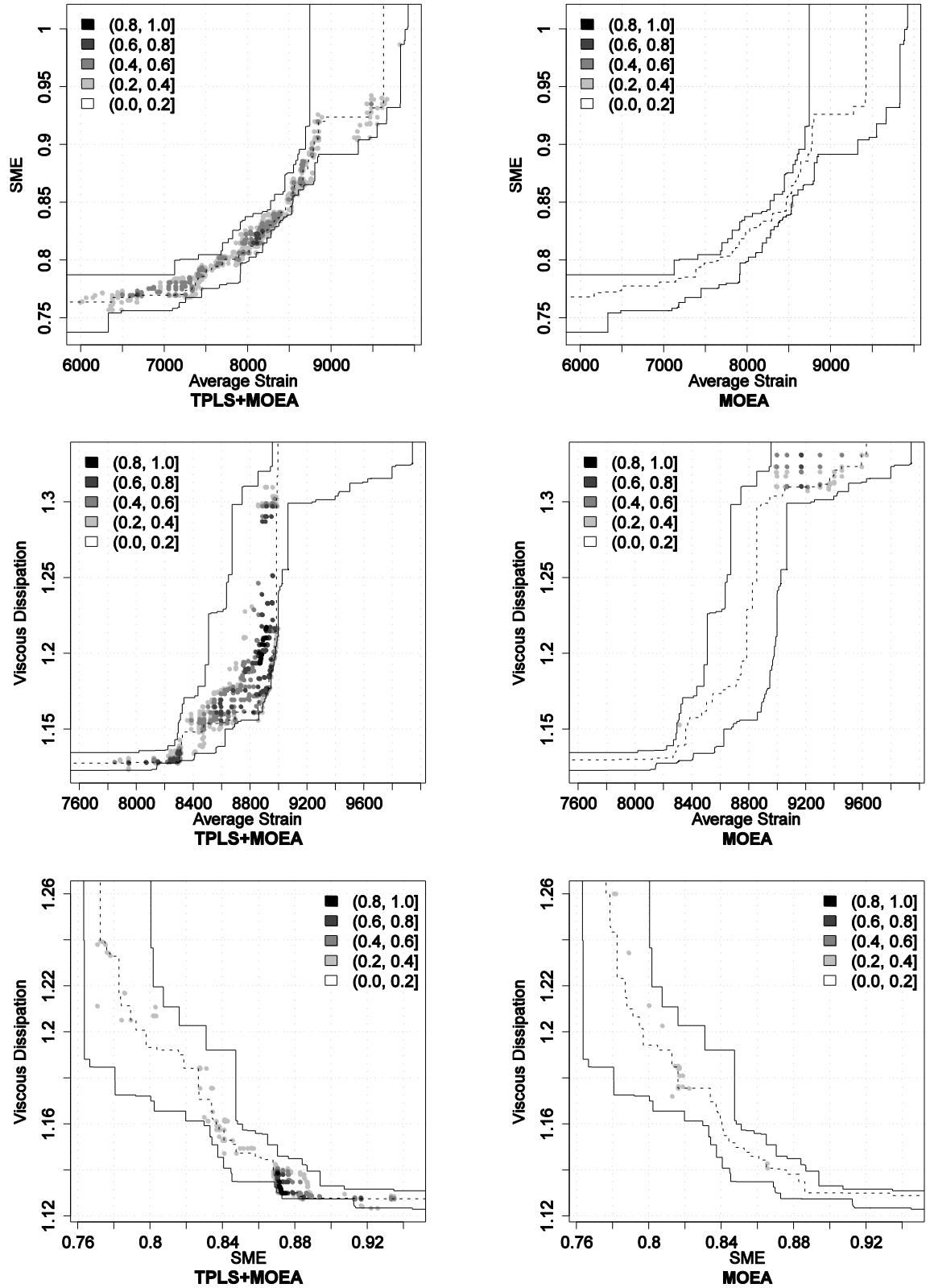

Figure 7.5. Results for instance TSCP4. Differences in terms of empirical attainment functions between the TPLS+MOEA algorithm and MOEA after 3000 evaluations of the simulation program. Advantages in favor of TPLS+MOEA are indicated on the left side; those in favor of MOEA on the right side.

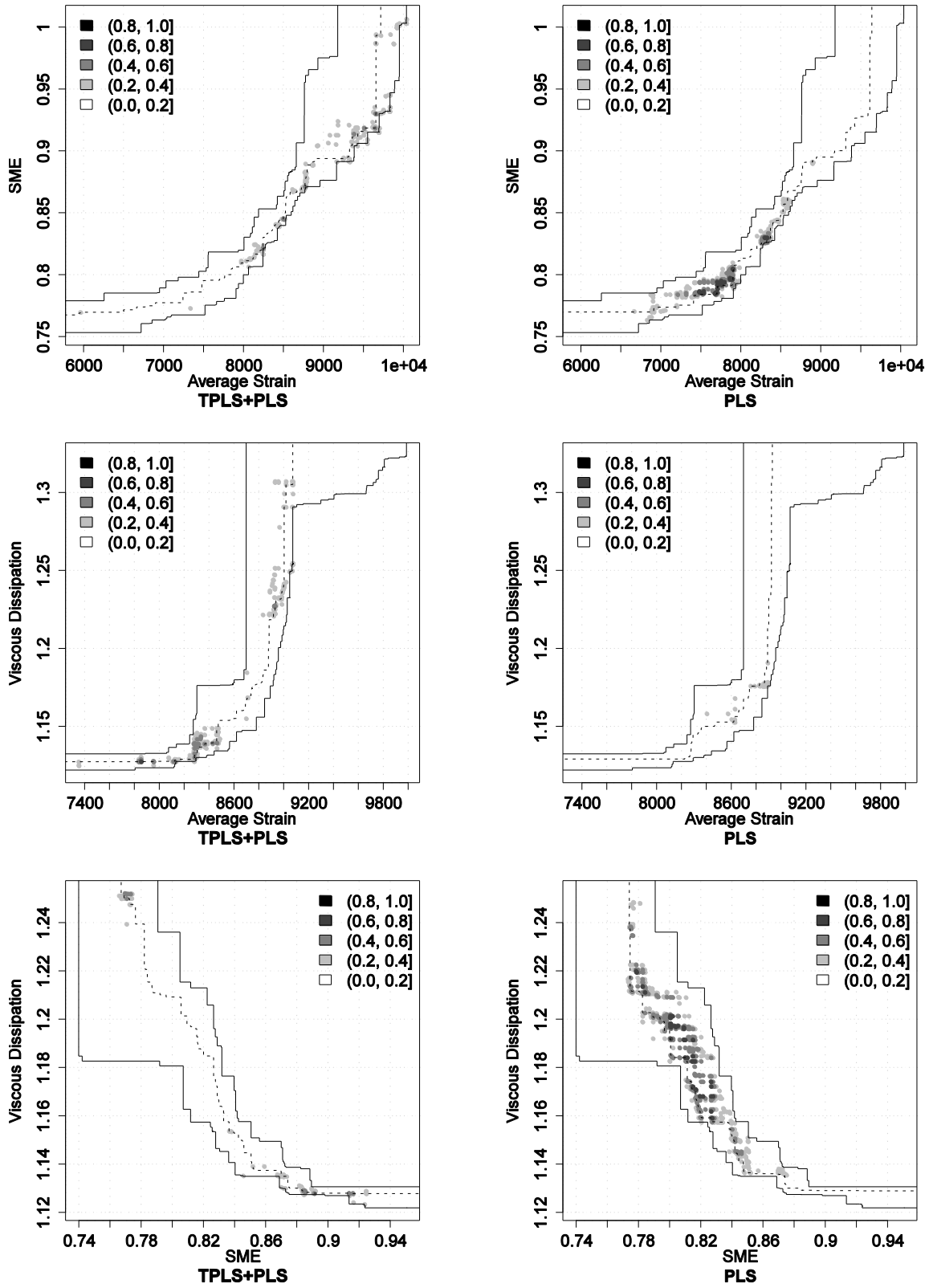

Figure 7.6. Results for instance TSCP4. Differences in terms of empirical attainment functions between the TPLS+PLS algorithm and PLS after 3000 evaluations of the simulation program. Advantages in favor of TPLS+PLS are indicated on the left side; those in favor of PLS on the right side.

7.3.3. Other comparisons

Finally, it may be interesting to compare the performance of the various hybrid algorithms developed. When comparing MOACO+PLS with MOEA+PLS, in general the advantages are on the side of MOACO+PLS algorithm (see Figure 7.7 and Appendix C, Figures C.19–C.21). For a small number of restrictive screw elements, TSCP1, this advantage is limited to small areas of the objective space (Figure C.19, Appendix C). However, the size of the areas and the intensity of the differences is higher for a higher number of restrictive screw elements. Similarly, the advantages are in favor of the TPLS+MOACO over the TPLS+MOEA algorithm as can be seen in Figure 7.8 and in Appendix C, Figures C.22–C.24.

When the two types of hybrid is compared, that is, MOACO+PLS and TPLS+MOACO or MOEA+PLS and TPLS+MOEA, the advantages are typically in favor of the hybrids that use a post-processing by PLS. In fact, in the case of the MOACO hybrids, the advantage of MOACO+PLS over TPLS+MOACO is rather large (Figure 7.9 and Appendix C, Figures C.25–C.27), while in the case of MOEA+PLS vs. TPLS+MOEA, the advantage of MOEA+PLS over TPLS+MOEA is restricted to some areas of the objective space (Figure 7.10 and Appendix C, Figures C.28–C.30).

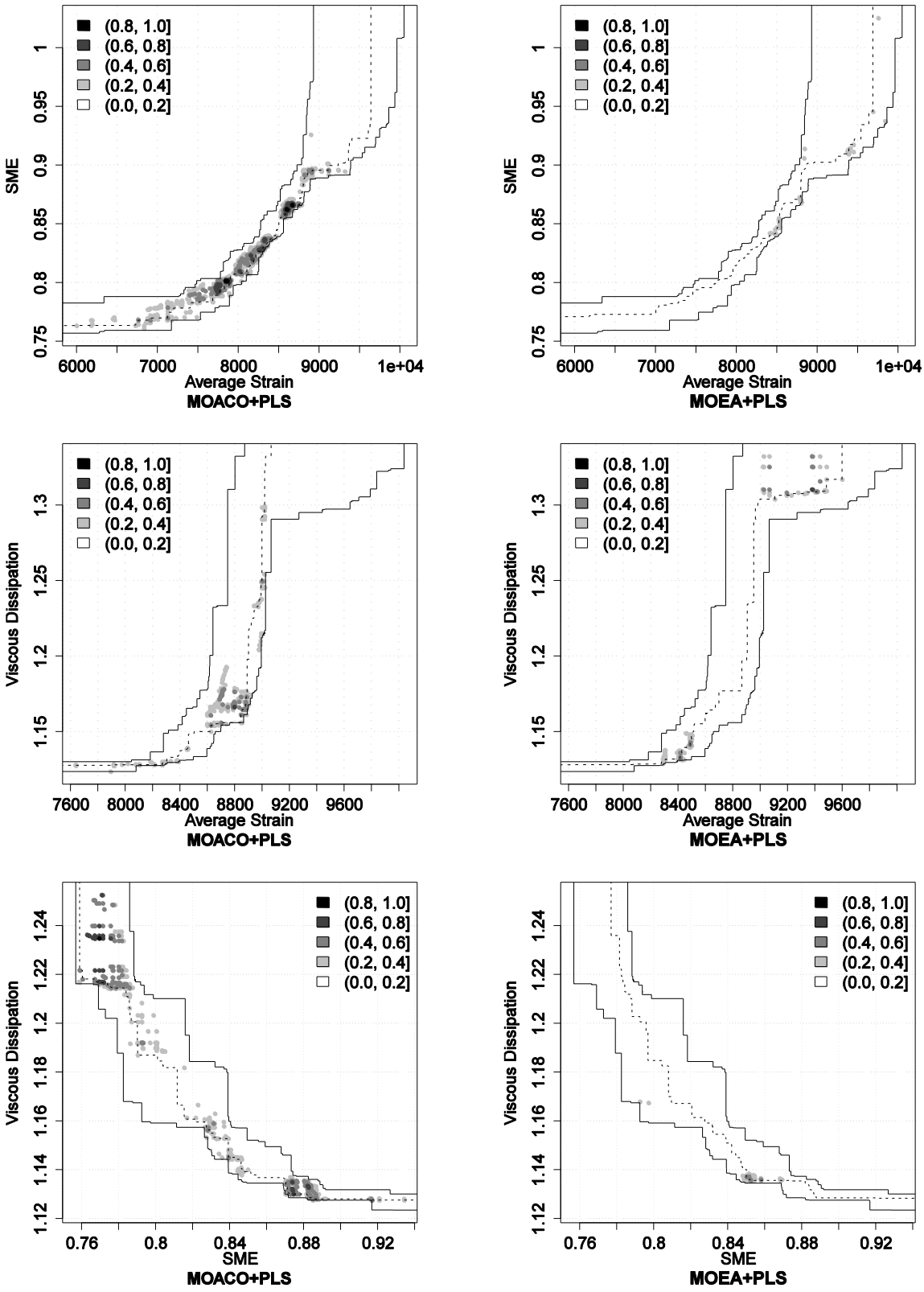

Figure 7.7. Results for instance TSCP4. Differences in terms of empirical attainment functions between the MOACO+PLS algorithm and MOEA+PLS after 3000 evaluations of the simulation program. Advantages in favor of MOACO+PLS are indicated on the left side; those in favor of MOEA+PLS on the right side.

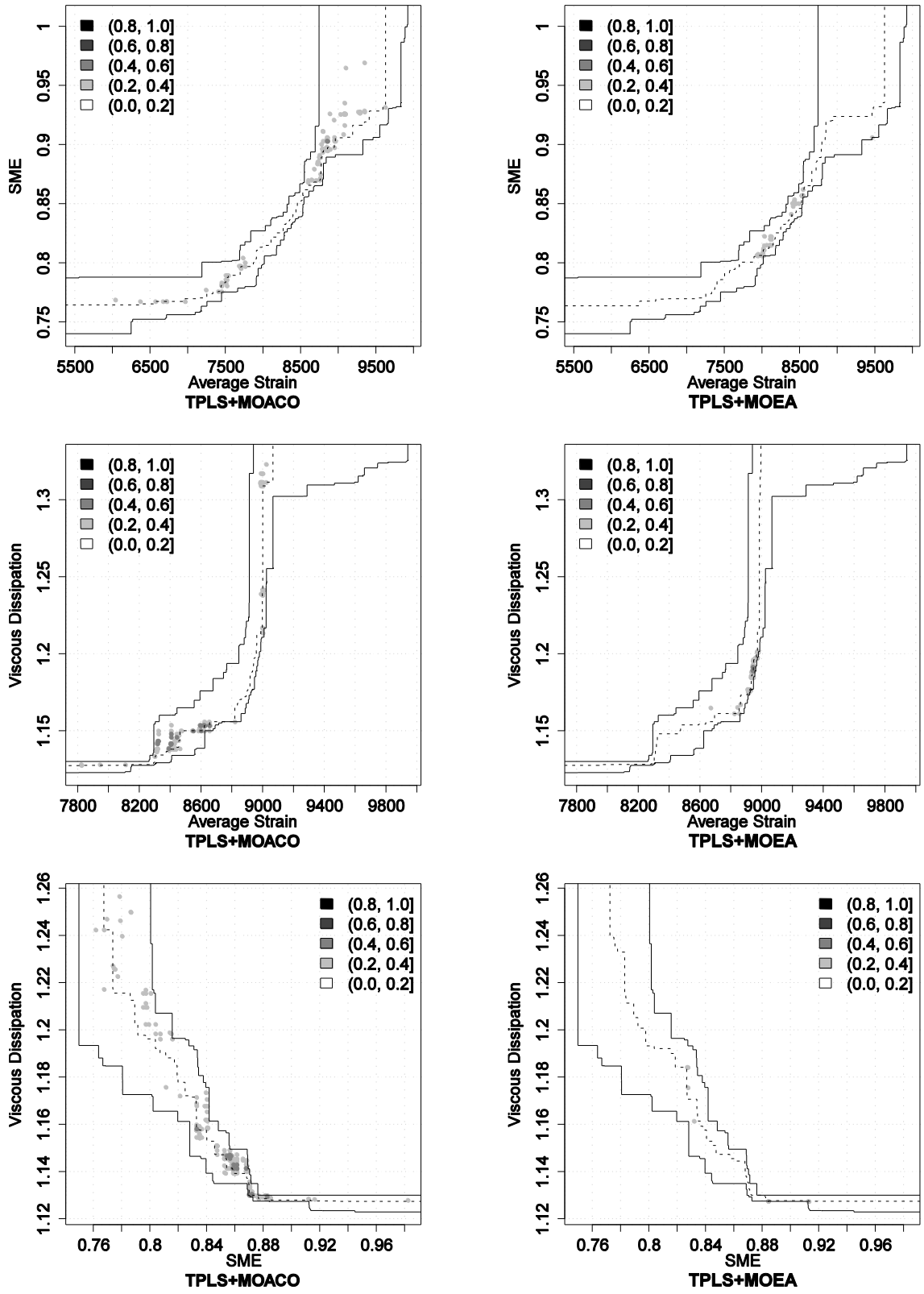

Figure 7.8. Results for instance TSCP4. Differences in terms of empirical attainment functions between the TPLS+MOACO algorithm and TPLS+MOEA after 3000 evaluations of the simulation program. Advantages in favor of TPLS+MOACO are indicated on the left side; those in favor of TPLS+MOEA on the right side.

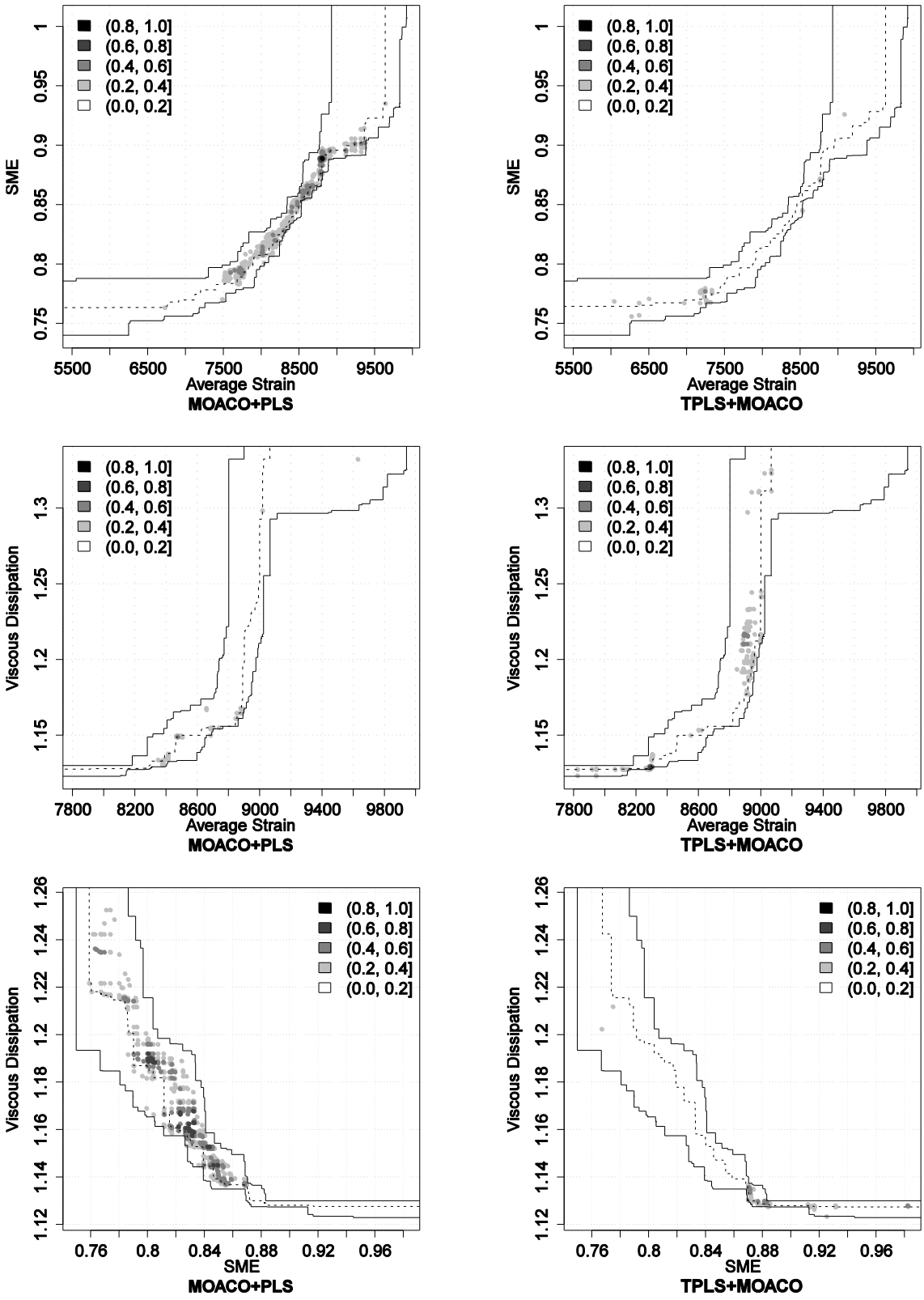

Figure 7.9. Results for instance TSCP4. Differences in terms of empirical attainment functions between the MOACO+PLS algorithm and TPLS+MOACO after 3000 evaluations of the simulation program. Advantages in favor of MOACO+PLS are indicated on the left side; those in favor of TPLS+MOACO on the right side.

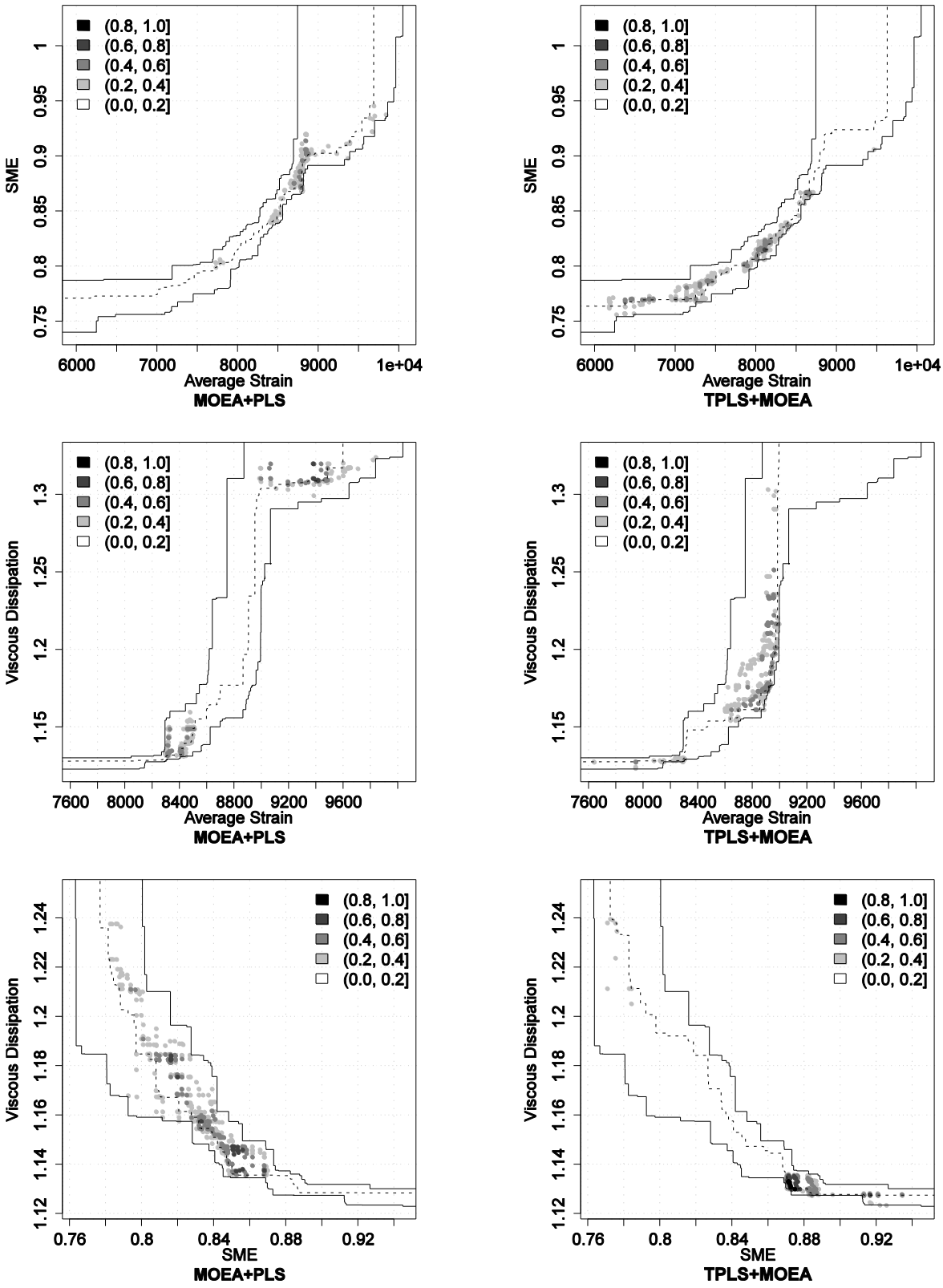

Figure 7.10. Results for instance TSCP4. Differences in terms of empirical attainment functions between the MOEA+PLS algorithm and TPLS+MOEA after 3000 evaluations of the simulation program. Advantages in favor of MOEA+PLS are indicated on the left side; those in favor of TPLS+MOEA on the right side.

7.3.4. Overall results and statistical analysis

As the comparison of the performance of the hybrid and non-hybrid through the EAF differences is not always fully conclusive, for each of the algorithms the hypervolume indicator is also computed. It measures for each algorithm run the area dominated by the non-dominated front with respect to a reference point. As mentioned above, for doing so the results of each algorithm were normalized taking into account the results obtained by all algorithms. Thus, for each algorithm 10 independent values for the achieved hypervolume on each case study was obtained.

In a first step, a statistical analysis of the so obtained results on each of the 12 case studies is performed. For the statistical analysis, the Friedman test is applied. Each algorithm is run using at each independent trial the same random number seed as a means to reduce the variance of the results; thus, the common random number seed also serves as the blocking factor for the Friedman test [CON99]. Then, the minimum significant difference of the sum of ranks for which a pair of results is considered significantly different (Post-Hoc Friedman test) is computed. For each of the twelve case studies, the results of this statistical analysis are given in Table 7.1.

Table 7.1. Results obtained after applying Friedman and Post-Hoc Friedman tests. For each case study, algorithms are ordered according to the rank obtained. The numbers in parenthesis are the difference of ranks relative to the best strategy. The strategy significantly better than the other ones is indicated in bold face.
The matching between the numbers in the ranking and the algorithms is as follows: 1.MOACO; 2. MOACO+PLS; 3. MOEA; 4.MOEA+PLS; 5.PLS; 6.TPLS; 7.TPLS+MOACO; 8.TPLS+MOEA; 9.TPLS+PLS. The notation used for the instances is as follows: the first number is the instance (1-TSCP1; 2-TSCP2; 3-TSCP3; 4-TSCP4) and the second number is the case study.

Instance	Rdiff	Ranking								
1 – 1	20.02	**4 (0)**	**2 (6)**	**1 (19)**	9 (30)	7 (32)	3 (34)	6 (37)	5 (38)	8 (56)
1 – 2	19.98	**2 (0)**	**4 (8)**	**9 (16)**	**1 (17)**	3 (21)	6 (21)	7 (27)	5 (38)	8 (59)
1 – 3	12.84	**7 (0)**	**2 (0.5)**	**1 (7)**	8 (22.5)	9 (41.5)	4 (42)	5 (46.5)	3 (50)	6 (64.5)
2 – 1	17.71	**2 (0)**	**4 (6)**	**1 (17)**	7 (18)	3 (38)	5 (44)	8 (45)	6 (50)	9 (52)
2 – 2	19.75	**2 (0)**	**4 (4)**	7 (23)	1 (26)	9 (31)	8 (42)	6 (43)	3 (44)	5 (48)
2 – 3	17.81	**2 (0)**	**7 (4)**	**1 (9)**	**3 (12)**	8 (19)	4 (24)	9 (34)	5 (48)	6 (57)
3 – 1	21.18	**2 (0)**	**7 (20)**	4 (22)	9 (23)	5 (24)	1 (25)	8 (42)	3 (46)	6 (50)
3 – 2	21.44	**2 (0)**	**1 (5)**	**4 (7)**	**9 (16)**	**5 (17)**	7 (22)	6 (31)	3 (39)	8 (43)
3 – 3	21.77	**2 (0)**	**9 (15)**	**5 (20)**	4 (23)	1 (28)	7 (30)	8 (38)	6 (41)	3 (48)
4 – 1	Inf	**4 (0)**	**5 (1)**	**9 (1)**	**7 (6)**	**2 (7)**	**8 (9)**	**1 (19)**	**6 (24)**	**3 (32)**
4 – 2	21.65	**7 (0)**	**2 (4)**	**4 (6)**	**8 (7)**	**9 (13)**	**1 (19)**	5 (28)	6 (33)	3 (43)
4 – 3	19.95	**2 (0)**	**4 (19)**	5 (21)	7 (23)	1 (25)	9 (38)	8 (39)	3 (47)	6 (58)

From these results we can observe that MOACO+PLS overall appears to be the best performing algorithm. In fact, for eight case studies is ranks best and it is never statistically significantly worse than the best ranking algorithm. As another high performing algorithm appears MOEA+PLS, which for nine case studies is among the top three ranking algorithms and never among the worst three on the others.

In a final attempt to even summarize more the obtained results, a table that summarizes the ranking of each of the algorithms across the case studies is computed; see Table 7.2 for details. These results further confirm the insights obtained by the discussion of the results based on the EAF differences plots and the ranking results in Table 7.1. In fact, a Friedman test on these additional data, using now the case studies as a blocking factor may also applied.

Table 7.2. For each algorithm the rank on each of the twelve case studies. Rank 1 refers to the best algorithm, rank 9 to the worst ranking algorithm.

Instance	MOACO	MOACO+PLS	MOEA	MOEA+PLS	PLS	TPLS	TPLS+MOACO	TPLS+MOEA	TPLS+PLS
1-1	3	2	6	1	8	7	5	9	4
1-2	4	1	5	2	8	6	7	9	3
1-3	3	2	8	6	7	9	1	4	5
2-1	3	1	5	2	6	8	4	7	9
2-2	4	1	8	2	9	7	3	6	5
2-3	3	1	4	6	8	9	2	5	7
3-1	6	1	9	3	5	8	2	7	4
3-2	2	1	8	3	5	7	6	9	4
3-3	5	1	9	4	3	8	6	7	2
4-1	7	5	9	1	2	8	4	6	3
4-2	6	2	9	3	7	8	1	4	5
4-3	5	1	8	2	3	9	4	7	6

In Table 7.3 we give the results of this Friedman test to give an indication of which are the top performing algorithms. In fact, MOACO+PLS is confirmed as the best performing algorithm for the TSCP across the twelve case studies. It is statistically significantly better than all other algorithms with the only exception being MOEA+PLS. In addition, these results also indicate that overall the hybrid algorithms typically rank better than the non-hybrid search algorithms on which they are based upon, thus, confirming the usefulness of hybridization in the algorithmic approaches to the TSCP.

Table 7.3. Results obtained after applying Friedman and Post-Hoc Friedman tests on the data from Table 7.2. Algorithms are ordered according to the rank obtained. The numbers in parenthesis are the difference of ranks relative to the best strategy. The strategy significantly better than the other ones is indicated in bold face.

Rdiff	**Ranking**								
17.83	**MOACO+PLS (0)**	**MOEA+PLS (16)**	TPLS+MOACO (26)	MOACO (32)	TPLS+PLS (38)	PLS (52)	TPLS+MOEA (61)	MOEA (68)	TPLS (76)

7.4. Conclusions

A main conclusion from the computational results is that the hybridization of the algorithms in the context of TSCP improves performance when compared to the main underlying, non-hybrid search methods. Though the improvements obtained through hybridization are not necessarily statistically significant in all cases, in many they are. Given the relatively small sample size of ten independent runs of each method on each case study and the resulting little power of the statistical tests, this result seems to indicate a rather general trend. The hybrid algorithms can be classified into two main approaches. The first is to use PLS as a post-processing of the non-dominated set of solutions generated by TPLS, MOACO, or MOEA. In this case, large improvements are rather clear obtained over the non-hybrid search methods used in the first phase for a same computational efforts: as shown in Section 7.3., the hybrid TPLS+PLS, MOACO+PLS and MOEA+PLS improve consistently and significantly over the underlying TPLS, MOACO and MOEA search strategies, respectively. The second main approach for hybridization was to seed the initial set of solutions to PLS, MOACO, and MOEA, respectively, by the non-dominated solutions returned by TPLS. In this case, TPLS was using only very few (in our case, four) weight vectors to limit its computation time.

While overall better ranking was observed for the hybrid algorithms, the relative improvements were generally smaller than observed for the first class of approaches. Hence, the experimental results clearly identify the post-processing of non-dominated sets by PLS as the most promising hybrid approach.

Overall, the result is that the MOACO+PLS hybrid is the best performing algorithm; it reaches the overall best ranking among the nine algorithms compared in this study and obtains on eight of the

twelve case studies the highest hypervolume. It ranks significantly better than the other competing algorithms with the only exception being MOEA+PLS. Hence, this study revealed that the hybrid algorithms are new state-of-the-art algorithms for the bi-objective twin-screw configuration problems we are faced with and their usage in the future are recommended.

There are a number of possible directions for future research on the TSCP. First, the performance of the algorithms can be improved by trying to introduce recent improvements on the anytime behavior especially of PLS [DUB12]. Such extensions may prove to be crucial for extending these algorithms to three and more objectives.

Finally, more advanced versions of the problem can be explored, where in a first step appropriate screw elements have to be chosen from a large set of available ones and then the chosen screw elements have to be sequenced in the second step. This more complex problem results, unfortunately, in much larger search spaces and, thus, it provides a significant challenge for future research.

Chapter 8

Optimization of Starch Cationization: A Practical Application*

The multi-objective optimization methodology based on Evolutionary Algorithms proposed in previous chapters is used in starch cationization case studies, involving different screw elements and screw lengths. The results obtained, taking into consideration various conflicting objectives, such as minimizing the specific mechanical energy and maximizing output and reaction efficiency, have physical meaning and thus validate the methodology.

---

* The contents of this chapter was adapted from: Teixeira, C., Covas, J.A., Berzin, F., Vergnes, B. and Gaspar-Cunha, A., 2011. Application of evolutionary algorithms to the definition of the optimal twin-screw extruder configuration for starch cationization. *Polymer Engineering and Science,* 51 (2), 330–340.

8.1. Introduction

In a series of previous articles, methods were developed to calculate and optimize starch cationization in a twin-screw extruder [BER07a, BER07b]. Starch cationization consists in replacing the hydroxyl groups of the starch backbone by quaternary ammonium salts to make these modified starches more appropriate for paper-making applications [TAR04]. This can be achieved in a reactive extrusion process, using a co-rotating twin-screw extruder, in which the reagent is injected after melting the starch. Starch cationization is quantified by the reaction efficiency (RE), calculated as the proportion of reagent having led to a substitution of hydroxyl group per anhydroglucose unit. The process has been modeled using the software Ludovic© [VER98], coupled to the appropriate kinetic equations [BER07a]. This modeling approach was also used for process optimization and scale-up, using a crude step by step method [BER07b]. In other papers [GAS02, GAS05], more sophisticated algorithmic optimization methods were used for optimizing both the processing conditions and the screw profile in twin-screw extrusion applications. In this chapter, these optimization techniques are applied to the starch cationization by reactive twin-screw extrusion. In fact, one of the key issues when using the twin-screw extruder for starch cationization concerns the definition of the best screw configuration yielding the best process performance (Twin Screw Configuration Problem).

This chapter is organized as follows. In Section 8.2, the methodology used for modeling flow and starch cationization is presented. The multi-objective optimization evolutionary algorithm used is described in Section 8.3. Then, in Section 8.4, the methodology is applied to a few case studies and the results obtained are discussed. The major conclusions are presented in Section 8.5.

8.2. Modeling Methodology

8.2.1. Flow Modeling

The efficiency of the chemical reaction occurring during the starch cationization process depends on some flow parameters, such as temperature and residence time. These parameters are strongly correlated with the operating conditions imposed on the extruder and with the screw configuration adopted. The screws usually comprise a number of screw elements of three distinct types: right-handed elements, left-handed elements, and blocks of staggered kneading discs. The geometrical

characteristic of each type promotes different flow features and, consequently, a different global performance. Each screw configuration can be evaluated by using software able to compute the evolution of the main flow parameters (average and cumulative residence time, pressure, power consumption, average strain, specific mechanical energy, viscosity, fill ratio, etc.) along the sequence of different screw elements. In previous studies [BER07a, BER07b, GAS02, GAS05], the Ludovic© software, developed at CEMEF, was used for this purpose [VER98]. In this chapter, the program TwiXtrud, recently developed at University of Minho will be also adopted, as it takes into account the different physical steps occurring along the extruder [TEI07]: (i) solids conveying without pressure; (ii) solids conveying under pressure; (iii) delay zone; (iv) melting (with high solids content); (v) melting (with low solids content); (vi) melt conveying under pressure, and (vii) melt conveying without pressure. The details about the modeling routine applied are presented in Chapter 3. The main differences between the Ludovic© software and the TwiXtrud program are due to the inclusion of solids conveying and melting by the latter. Moreover, a temperature profile across the channel is considered, whereas Ludovic© computes an average temperature.

8.2.2. Starch cationization modeling

A detailed discussion on starch cationization modeling can be found in reference [BER07a]. The reaction is assumed to follow a second order kinetics, depending on time, temperature and reagent concentration.

$$\underset{\text{Starch}}{A} + \underset{\text{Reagent}}{B} \xrightarrow{k} \underset{\text{Cationic starch}}{C} \tag{8.1}$$

The rate of consumption of the OH starch reacting groups equals the rate of consumption of reagent B and the rate of formation of starch cationic groups:

$$-\frac{d[A]_t}{dt} = -\frac{d[B]_t}{dt} = \frac{d[C]_t}{dt} = k[A]_t[B]_t \tag{8.2}$$

The concentration of starch hydroxyl groups at time t, $[A]_t$ can be expressed as:

$$[A]_t = \frac{\frac{[A]_0}{[B]_0}\left([A]_0 - [B]_0\right)\exp\left[\, k\left([A]_0 - [B]_0\right)t\right]}{\frac{[A]_0}{[B]_0}\exp\left[\, k\left([A]_0 - [B]_0\right)t\right] - 1} \tag{8.3}$$

where $[A]_0$ and $[B]_0$ are the initial concentrations in starch and reagent.

Defining the degree of substitution as the number of cationic groups linked per anhydroglucose unit (one anhydroglucose unit bears three OH groups), the theoretical maximum degree of substitution is:

$$DS_{th} = 3\frac{[B]_0}{[A]_0} \tag{8.4}$$

From the concentration of starch hydroxyl groups at time t, $[A]_t$, the degree of substitution can be expressed as:

$$DS = \frac{DS_{th}}{[B]_0}\left([A]_0 - [A]_t\right) \tag{8.5}$$

and the reaction efficiency RE is defined as:

$$RE = \frac{DS}{DS_{th}} \tag{8.6}$$

It has been demonstrated that the development of the cationization reaction does not modify the starch rheological behavior [BER07c]. Consequently, no coupling between flow conditions and reaction progress is necessary. In the present study, the same kinetics has been used as in reference [BER07a], but utilizing the TwiXtrud flow model. Anyway, in order to assure that the

modeling results produced by TwiXtrud can be used for optimization purposes, it is first necessary to be certain that both flow models yield similar results.

8.2.3. Comparison between Modeling Software

Ludovic© and TwiXtrud were used to calculate the flow for different screw speeds and feed rates for the same screw profile. Figure 8.1 compares the corresponding predictions of the evolution along the extruder of temperature, pressure and cumulative residence time.

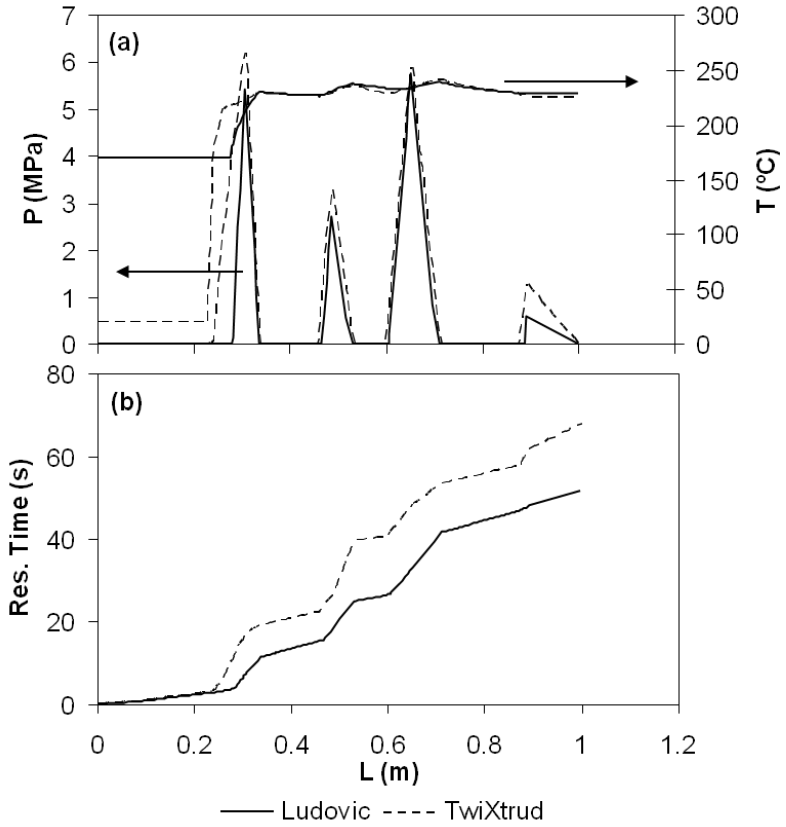

Figure 8.1. Predicted melt temperature, pressure and average residence time axial profiles by Ludovic© and TwiXtrud (processing a polypropylene at 8 kg/h, with a screw speed of 150 rpm, barrel set at 220 °C and using the following screw configuration: RH/97.5/45; RH/150/30; RH/60/20; LH/30/-30; RH/30/30; RH/120/30; KB/45/-30°; RH/60/45; RH/60/30; KB/60/-60°; RH/150/30; RH/30/20).

As expected, the main differences arise for the melting stage. Since the TwiXtrud program takes melting into consideration, the calculations start earlier, with a more important fill ratio. As a consequence, the temperature increases more upstream and the mean residence time is slightly higher, which may be important when reactive systems are considered.

If the comparison is generalized to different screw speeds and feed rates, the results concerning residence time and temperature are generally in close agreement with experiments (data taken from [BER07a]). As for the reaction efficiency, Figures 8.2 and 8.3 show that both codes provide similar tendencies, even if Ludovic© is closer to the experimental values.

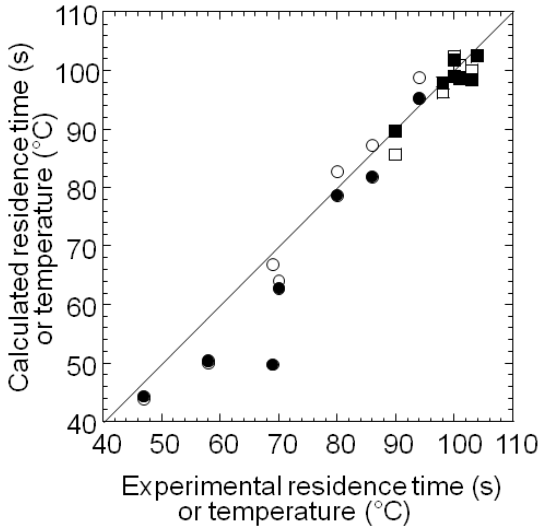

Figure 8.2. Calculated results from Ludovic© (open symbols) and TwiXtrud (filled symbols) and equivalent experimental data (circle: residence time, square: temperature).

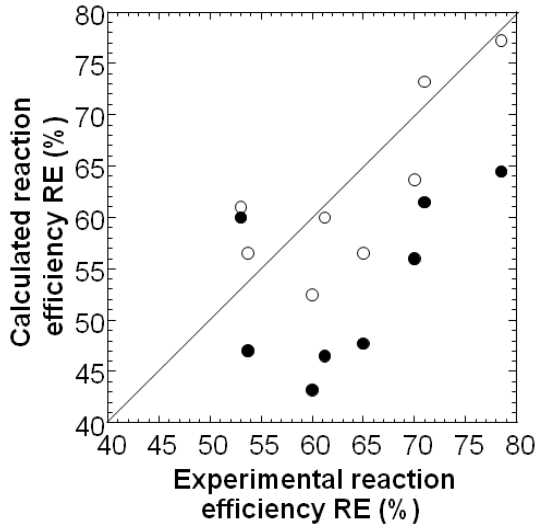

Figure 8.3. Calculated results from Ludovic© (open symbols) and TwiXtrud (filled symbols) and equivalent experimental data (reaction efficiency).

Figure 8.4 compares the RE predictions when using the 5 screw profiles that will be selected for the case studies to be discussed in the optimization section (Table 8.1), for 5 different outputs ranging between 2.5 and 40 kg/h.

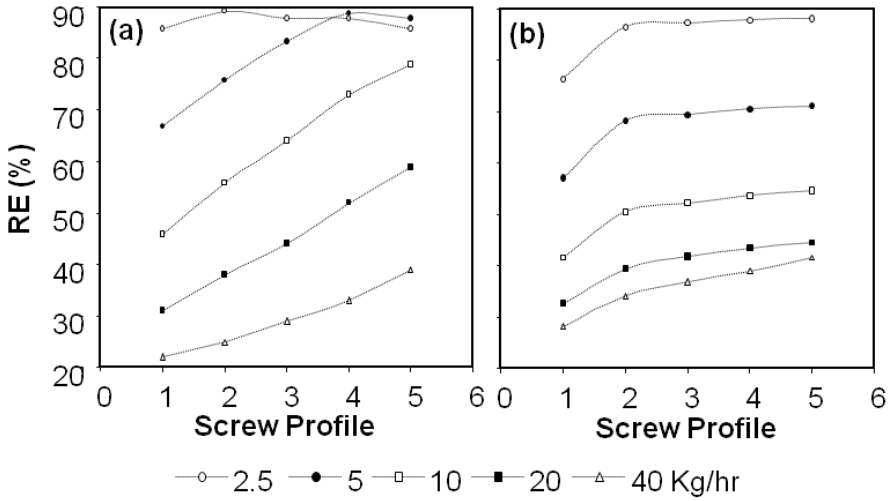

Figure 8.4. Influence of output and screw configuration on the reaction efficiency, RE: (a) Ludovic©; (b) TwiXtrud.

Table 8.1. Individual screw elements for 5 different screw profiles (*L* and *P* are the length and the pitch of each element, respectively). KB indicates a block of 5 kneading discs, with a staggering angle of -45°. LH indicates a left-handed screw element. Elements are ranked from hopper to die.

Profile	Screw Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	*L* (mm)	250	50	50	50	50	50	50	50	50	125	125							
	P (mm)	33.3	25	16.6	KB -45°	33.3	25	16.6	KB -45°	33.3	25	16.6							
2	*L* (mm)	250	50	50	50	50	50	50	50	50	50	50	50	50					
	P (mm)	33.3	25	16.6	KB -45°	33.3	25	16.6	KB -45°	33.3	25	16.6	KB -45°	25	16.6				
3	*L* (mm)	250	50	50	50	50	50	50	50	50	50	50	50	25	25	25	25		
	P (mm)	33.3	25	16.6	KB -45°	33.3	25	16.6	KB -45°	33.3	25	16.6	KB -45°	25	16.6	LH 16.6	16.6		
4	*L* (mm)	250	50	50	50	50	50	50	50	50	25	25	25	25	50	25	25	25	25
	P (mm)	33.3	25	16.6	KB -45°	33.3	25	16.6	KB -45°	33.3	25	LH 16.6	25	16.6	KB -45°	25	16.6	LH 12.5	16.6
5	*L* (mm)	250	50	50	50	50	25	25	50	50	50	25	25	25	25	50	25	25	25	25
	P (mm)	33.3	25	16.6	KB -45°	33.3	25	LH 16.6	16.6	KB -45°	33.3	25	LH 16.6	25	16.6	KB -45°	25	16.6	LH 12.5	16.6

The general trends provided by the two computations are similar: for a fixed screw profile, RE increases when the feed rate decreases; for a fixed feed rate, RE increases with the restrictive character of the screw profile (from 1 to 5). The values of the calculated RE are also very close, varying from 20 to 90%, even if the results obtained with TwiXtrud fall below those of Ludovic©, as shown in Figure 8.3. However, differences are also noticeable. With Ludovic©, except at 2.5 kg/h, RE increases regularly when changing from screw profile 1 to screw 5. In the case of TwiXtrud, RE also increases but levels off rapidly, no significant difference existing for profiles 2 and 5. As

stressed before, this is probably due to the different temperature calculations performed by the two models (average temperature for Ludovic©, temperature profile for TwiXtrud). Therefore, and despite the small differences, these results show that TwiXtrud can be used for the optimization procedure.

8.3. Multi-Objective Optimization

TSCP can be seen as a sequencing problem, since a number of resources (in this case, these are the screw elements) must be allocated sequentially. Simultaneously, as different and eventually conflicting objectives must be satisfied, a Multi-Objective Combinatorial Problem (MOCP) must be solved.

Evolutionary Algorithms (EAs) are particularly adequate to tackle multi-objective optimization problems, since they work with a population of solutions that can evolve globally in the direction of the optimal Pareto front [DEB01]. In this work, the Reduced Pareto Set Genetic Algorithm (RPSGA) is used [GAS04].

The formulation of the TSCP as a sequencing problem implies that each solution comprises a sequence of discrete numbers representing the positions of the screw elements in the screw axis and also the use of the inver-over reproduction operator [GAS05, TAO98]. The existing RPSGA was modified accordingly, i.e., the real decision variables were replaced by this sequence of discrete numbers and the usual crossover and mutation operators were replaced by the inver-over operator.

The main steps of the RPSGA algorithm used in this work are illustrated in Algorithm 4.3. Details about this algorithm can be found in Subsections 4.4.2 and 5.3.1. See also [GAS04, GAS05].

8.4. Optimization Of Starch Cationization

8.4.1. Case Studies

A laboratory scale co-rotating twin screw extruder Clextral BC21 (Clextral, Firminy, France) is considered, having a screw diameter of 25 mm and a total length of 900 mm (length to diameter ratio L/D = 36). Table 8.1 presents the individual screw elements for 5 selected screw profiles. These profiles were used in a previous paper [BER07b] where the starch cationization process was optimized by a trial-and-error procedure. During the TSCP resolution, the

shadowed elements will be maintained in their original location. The reagent is injected at the end of the second element, i.e., at an axial distance of 300 mm. At the opposite screw end, since conveying elements are needed to create the required pressure for die flow, the last screw element is assumed as having fixed screw geometry, even if with a different length, depending on the screw geometry upstream. The restrictive character of the screws increases from profile 1 to 5.

Four different case studies were analyzed as illustrated in Table 8.2, which identifies the parameters to be optimized, the optimization objectives, the aim and range of variation of the objectives and the restrictions applied. Case study 1 aims at comparing the optimization strategy proposed here with the trial-and-error methodology used previously [BER07b]. Thus, only the output was considered, the objectives being to maximize both output and reaction efficiency (RE). In case study 2, the screw profile 5 (Table 8.1) was optimized for different output values, i.e., the best location of the different screw elements is defined using the RPSGA. Finally, case studies 3 and 4 tackle both screw configuration and output optimization. The former involves the minimization of the specific mechanical energy (SME), while the latter aims at maximizing output. In all runs, the screw speed N, the barrel temperature T_b, the reagent injection point and amount were kept constant and equal to $N = 400\text{rpm}$, $T_b = 130^{\circ}$, 300 mm, and 0.107 Q_{starch}, where Q_{starch} is the starch feed rate, respectively. As explained previously [BER07a], the concentration of reagent is expressed as a theoretical degree of substitution (DS_{th}). As a starch glycosil unit has three hydroxyl groups, the maximum degree of substitution DS is 3. Although cationic starches used in paper industry usually have DS in the range 0.02-0.05, the value of 0.1 for DS_{th} was selected, as this is more difficult to reach and, henceforth, the optimization exercise is more interesting. The optimizations performed in all case studies used TwiXtrud as modeling program.

The main and elitist populations had 100 and 200 individuals, respectively, 50 generations having been studied. A roulette wheel selection strategy, an inver-over probability of 0.8, 30 ranks and limits of indifference of the clustering technique all equal to 0.01 were chosen (see [GAS04] for more details).

Table 8.2. Optimization runs

	RUNS	Screw Profile	Output	Parameter	Objectives	Aim	X_{min}	X_{max}	Restrictions
Case study 1	01	**1**		Output (kg/h)	RE (%)	Max	0	100	T_{Max} < 165°C
	02	**2**							
	03	**3**			Output (kg/h)	Max	2.5	40	SME < 0.72 MJ/kg
	04	**4**							
	05	**5**							
Case study 2	06	5	**2.5**	Screw Configuration (*)	RE (%)	Max	0	100	T_{Max} <165 °C
	07	5	**5**						
	08	5	**10**		SME (MJ/kg)	Min	0	1.5	
	09	5	**20**						
	10	5	**40**						
Case study 3	11	**1**		Screw Configuration (*) and Output (kg/h)	RE (%)	Max	0	100	T_{Max} <165 °C
	12	**2**							
	13	**3**			SME (MJ/kg)	Min	0	0.72	
	14	**4**							
	15	**5**							
Case study 4	16	**1**		Screw Configuration (*) and Output (kg/h)	RE (%)	Max	0	100	T_{Max} < 165°C
	17	**2**							
	18	**3**			Output (kg/h)	Max	2.5	40	SME < 0.72 MJ/kg
	19	**4**							
	20	**5**							

* The first two and the last screw elements (Table 8.1) were not changed.

8.4.2. Results and Discussion

Before discussing in detail each case study, it seems relevant to show how the optimization results obtained through a MOEA can be linked to the real problem under study, as well as to clarify the information contained in those results. Run 6 (Table 8.2) will be taken as an example, where the configuration of screw no. 5 (Table 8.1) is to be optimized in order to simultaneously maximize RE and minimize SME.

Figure 8.5 shows the solutions in the random initial population and the non-dominated solutions of the last population (at the 30th generation) represented on the objectives domain, i.e., the Pareto-front. Each of these solutions corresponds to a different screw profile that was evaluated by the modeling routine. The Pareto frontiers embody the trade-off between the different objectives. In this case the two objectives are conflicting, since when the RE, which is to be maximized, increases (improves), SME, which is to be minimized, increases as well (deteriorates). Throughout the

optimization the population evolves progressively towards a better performance (top left corner of the graph).

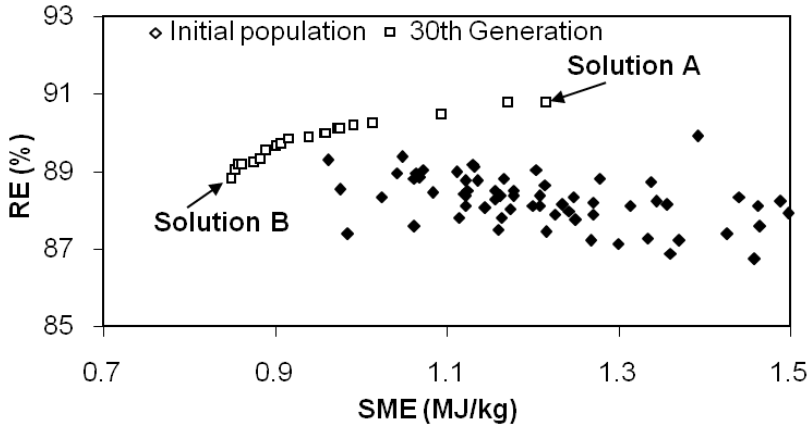

Figure 8.5. Random initial solutions and non-dominated solutions of the 30th generation.

Table 8.3 presents the two solutions obtained (i.e., screw configurations) for which RE is higher (solution A) and SME is lower (solution B), respectively. The temperature, residence time, RE and SME profiles along the extruder for these two solutions are shown in Figure 8.6.

Table 8.3. Optimal screw configurations for run 6 (solution A: maximum of RE, solution B: minimum of SME; *L* and *P* in mm).

Sol.		**Position**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
A	*L*	250	50	25	25	50	25	50	25	25	50	50	50	25	25	25	50	50	25	25
	P	33.3	25	25	LH 12.5	33.3	16.6	16.6	16.6	25	16.6	33.3	KB -45°	25	25	LH 12.5	KB -45°	KB -45°	LH 16.6	16.6
B	*L*	250	50	25	25	25	50	50	50	25	50	25	25	25	25	50	50	25	50	25
	P	33.3	25	LH 12.5	25	25	33.3	33.3	16.6	25	16.6	16.6	LH 16.6	25	LH 12.5	KB -45°	KB -45°	16.6	KB -45°	16.6

The best screw to maximize RE (solution A) has a restrictive element at position 2, which seems appropriate since in this case the reaction starts in the section located at position 1. Identical evidence cannot be directly inferred in the case of the screw that minimizes SME (solution B). In fact, one would expect the best screw to be the one with all the restrictive elements located as

downstream as possible. However, the optimization proposes a screw with a restrictive element in the first available location (location 1).

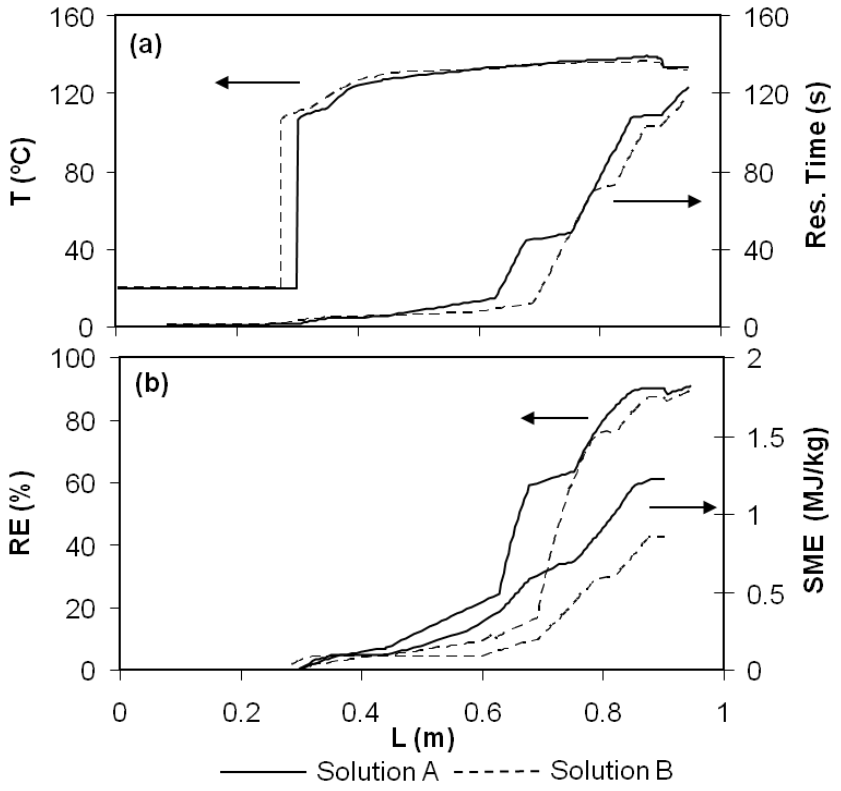

Figure 8.6. Modeling results for solutions A and B (Table 8.3).

Since SME is directly related to viscosity, this profile will ensure rapid melting and, consequently, a melt temperature as high as possible all along the screw. Moreover, the elements inducing high shear rates (left-handed and kneading discs) should be located where the temperature is the highest, i.e. near to the screw tip. In order to verify the value of this explanation, the first left-handed element of solution B, initially located at position 1 (see Table 8.3), was shifted to location 6 (denoted as solution B*) and the corresponding modeling results were compared with those for solution B (Figure 8.7). This confirms that solution B is indeed the best one, because solution B* produces a higher viscosity due to the lower melt temperature. Therefore, the optimization algorithm seems to be capable of taking into account the physical aspects of each one of these two objectives.

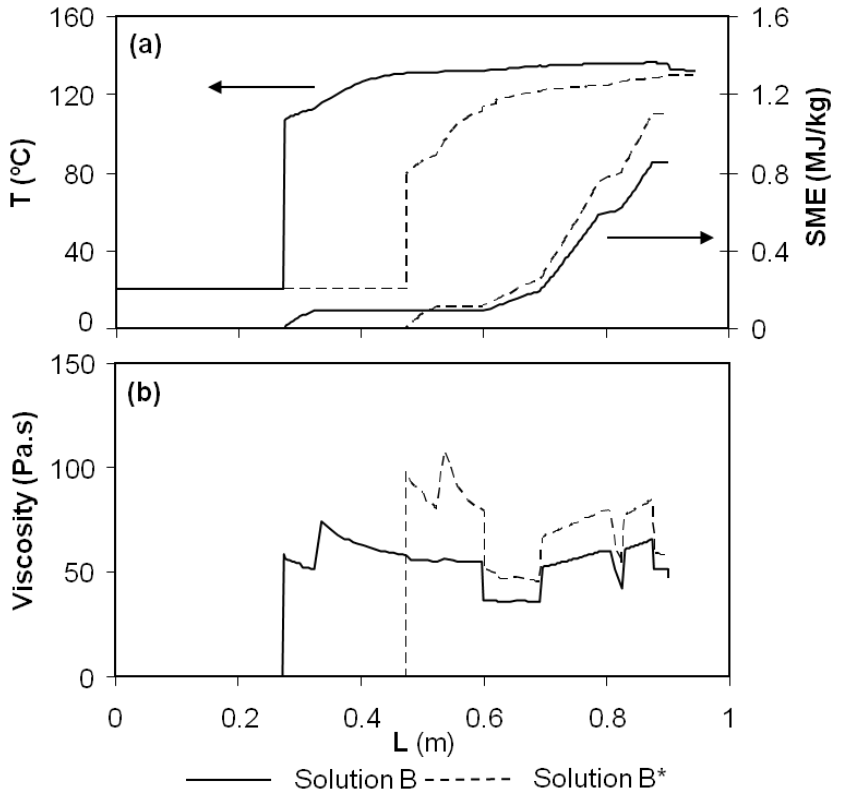

Figure 8.7. Modeling results for solutions B (Table 3) and B*.

Case study 1 can be used to illustrate the differences between the current optimization scheme and the previous effort [BER07b]. Here, an approximation to the optimal Pareto frontier (i.e., output versus RE) is generated automatically, taking simultaneously into consideration the objectives and the limiting temperature ($T_{\max} < 165$°C), specific mechanical energy ($SME < 0.72$MJ/Kg) and torque ($C < 50$Nm) values. The previous method consisted in the progressive definition (in a graph such as that represented in Figure 8.4) of feasibility contours by introducing the above limiting values one by one. Thus, a viable operating window, not an optimal set of solutions, is defined. Moreover, this method will not search for the best combination of screw configuration (which is constant in Figure 8.4) and operating conditions.

The various case studies will now be discussed in more detail. Figure 8.8 shows the optimal Pareto frontiers (output versus RE) for case study 1 (runs 1 to 5), where the aim is to maximize output and RE for the 5 screws, using the pre-set screw configuration. As anticipated, these two objectives are conflicting, i.e., they have opposite trends and, consequently, the Pareto frontiers have a convex shape, as well as an inflexion point for the higher output values.

Since upon progressing from profile 1 to 5 the severity of the screw profile increases (see Table 8.1), both temperature generation and residence time should increase and, consequently, so should do the extension of the reaction (i.e., RE). This can be indeed being observed on the graph containing the 5 optimal Pareto frontiers.

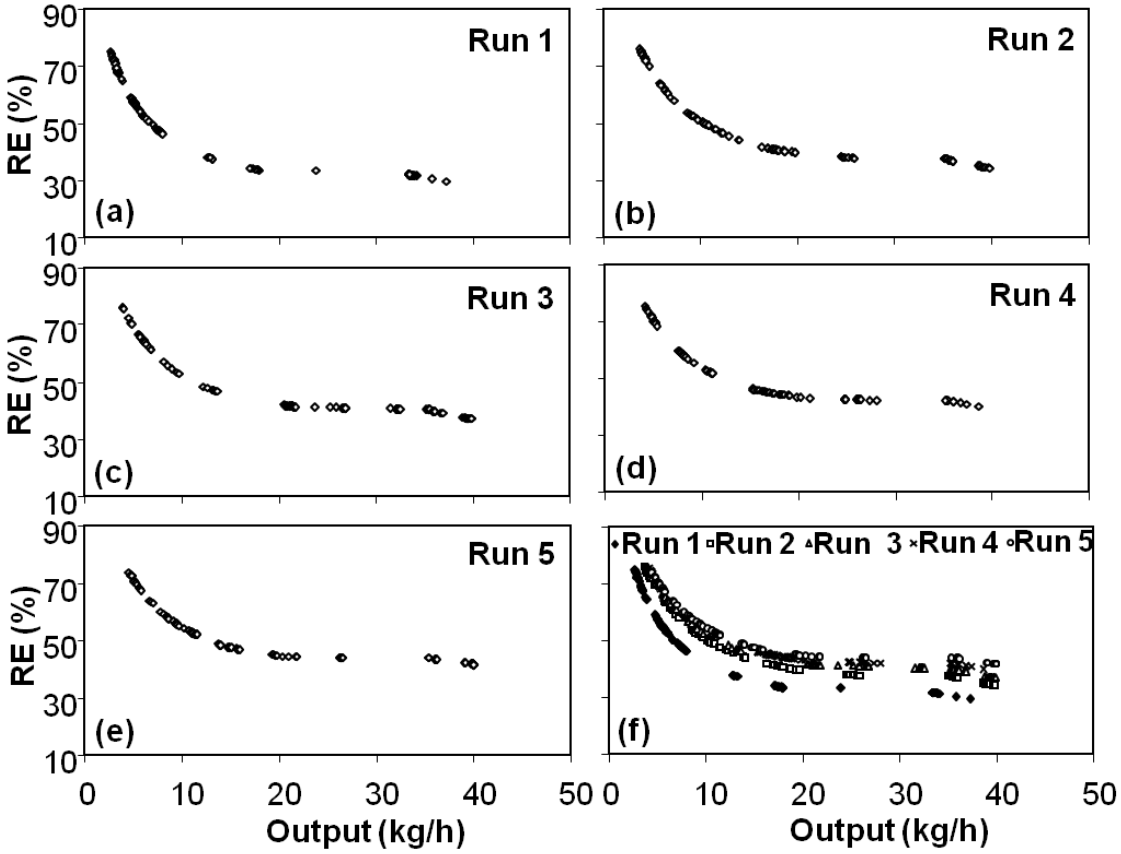

Figure 8.8. Optimal Pareto frontiers for case study 1 (runs 1 to 5, Table 7.2).

Figure 8.9 shows the optimal Pareto frontiers for case study 2 (runs 6 to 10), where the screw profile no. 5 in Table 8.1 was optimized in terms of maximizing RE and minimizing SME for different feed rates. It appears that the latter has little influence on the outcome, as the different runs are almost superimposed. As expected, the highest REs are obtained at the lowest feed rates, when the residence time is longer. For each run, two screws were selected from the Pareto curves, the one that maximizes RE and the one that minimizes SME (see Table 8.4). Given the results plotted in Figure 8.9, it is not surprising that the screws are similar for the different feed rates, but differ depending on the objective of that optimization: to maximize RE, the restrictive elements (kneading blocks and left-handed elements) are concentrated downstream, close to the screw tip. To minimize SME, these elements are arranged in series of two, separated by conveying elements.

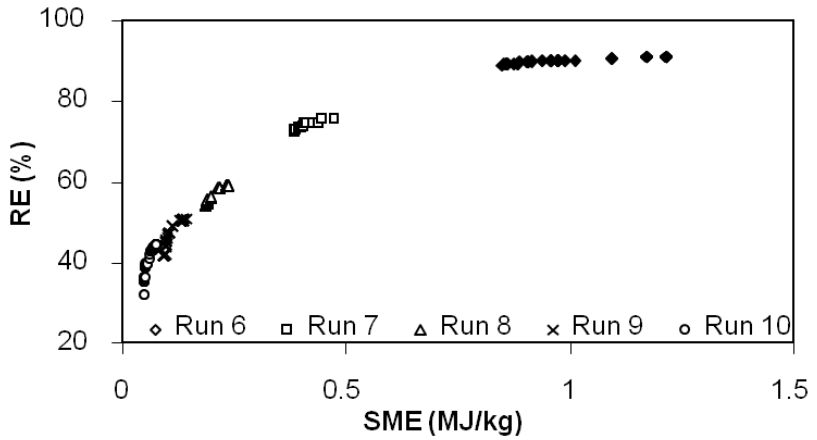

Figure 8.9. Optimal Pareto frontiers for case study 2 (runs 6 to 10, Table 8.2).

Similar results were obtained for case study 3 (Figure 8.10), where the screw configurations 1 to 5 and the feed rate were optimized simultaneously in order to maximize RE and minimize SME. Runs 16 to 20 correspond to screw profiles 1 to 5, respectively. As in Figure 8.8, Pareto frontiers move towards higher RE when the restrictive character of the profile increases. Also, the objectives are again contradictory: a low SME is obtained at high feed rates (shorter residence times), but at the cost of a low RE. The analysis of Table 8.5, which identifies the profiles of the optimized screws, reveals that the optimized conditions are reached for a feed rate in the ranges 2.5-3 kg/h or 37.2-39.9 kg/h, for maximizing RE and minimizing SME, respectively. Also, the optimized configurations are similar for the two objectives, i.e., after a rapid melting imposed by the presence of a left-handed element (when available), all the remaining restrictive elements are located near the screw tip.

Table 8.4. Optimized screw profiles for case study 2 (*L* and *P* in mm).

Run 06: 2.5 kg/hr

		Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
Max. RE	*L*	250	50	25	25	50	25	50	25	25	50	50	50	25	25	25	50	50	25	25
	P	33.3	25	25	LH 12.5	33.3	16.6	16.6	16.6	25	16.6	33.3	KB -45°	25	25	LH 12.5	KB -45°	KB -45°	LH 16.6	16.6
Min. SME	*L*	250	50	25	25	25	50	50	50	25	50	25	25	25	25	50	50	25	50	25
	P	33.3	25	LH 12.5	25	25	33.3	33.3	16.6	25	16.6	16.6	LH 16.6	25	LH 12.5	KB -45°	KB -45°	16.6	KB -45°	16.6

Run 07: 5.0 kg/hr

		Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
Max. RE	*L*	250	50	25	25	25	25	50	50	25	25	50	50	25	50	25	50	50	25	25
	P	33.3	25	16.6	LH 12.5	25	25	16.6	33.3	25	LH 16.6	33.3	16.6	16.6	KB -45°	25	KB -45°	KB -45°	LH 12.5	16.6
Min. SME	*L*	250	50	25	25	25	50	25	50	50	25	50	25	25	50	50	25	25	50	25
	P	33.3	25	16.6	LH 16.6	25	33.3	25	16.6	33.3	25	16.6	LH 12.5	16.6	KB -45°	KB -45°	25	LH 12.5	KB -45°	16.6

Run 08: 10.0 kg/hr

		Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
Max. RE	*L*	250	50	25	50	25	25	25	25	50	50	25	50	25	25	25	50	50	50	25
	P	33.3	25	LH 12.5	16.6	16.6	25	25	25	33.3	16.6	16.6	33.3	LH 16.6	25	LH 12.5	KB -45°	KB -45°	KB -45°	16.6
Min. SME	*L*	250	50	25	25	50	25	25	50	50	25	50	25	25	25	50	25	50	50	25
	P	33.3	25	LH 12.5	25	33.3	25	25	33.3	16.6	25	16.6	LH 12.5	16.6	LH 16.6	KB -45°	16.6	KB -45°	KB -45°	16.6

Run 09: 20 kg/hr

		Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
Max. RE	*L*	250	50	25	25	50	50	25	25	50	25	25	50	25	50	50	25	25	50	25
	P	33.3	25	25	LH 16.6	33.3	16.6	25	16.6	16.6	25	25	KB -45°	16.6	33.3	KB -45°	LH 12.5	LH 12.5	KB -45°	16.6
Min. SME	*L*	250	50	25	25	25	25	50	50	50	25	50	50	25	25	25	25	50	50	25
	P	33.3	25	16.6	LH 16.6	25	25	33.3	16.6	33.3	25	16.6	KB -45°	16.6	LH 12.5	LH 12.5	25	KB -45°	KB -45°	16.6

Run 10: 40 kg/hr

		Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	
Max. RE	*L*	250	50	25	50	25	50	25	50	50	25	25	50	50	25	25	25	50	25	25
	P	33.3	25	LH 16.6	33.3	25	33.3	16.6	16.6	KB -45°	25	25	KB -45°	16.6	16.6	25	LH 12.5	KB -45°	LH 12.5	16.6
Min. SME	*L*	250	50	25	50	50	25	50	25	25	25	25	50	50	50	25	25	50	25	25
	P	33.3	25	LH 16.6	33.3	33.3	16.6	16.6	25	25	25	25	KB -45°	16.6	KB -45°	LH 12.5	16.6	KB -45°	LH 12.5	16.6

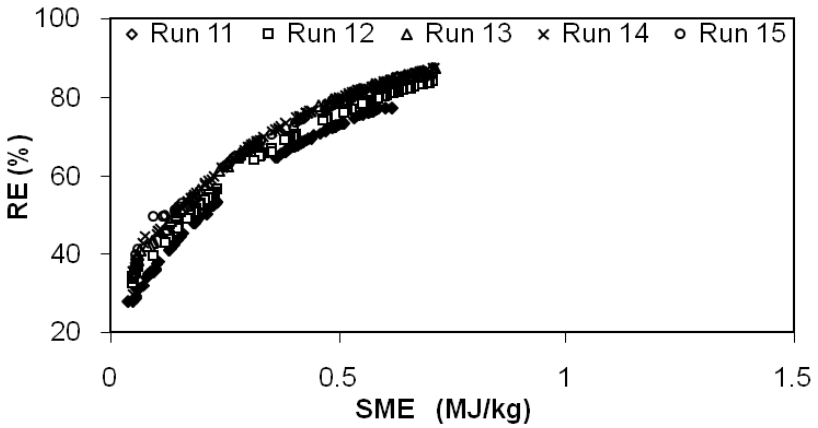

Figure 8.10. Optimal Pareto frontiers for case study 3 (runs 11 to 15, Table 7.2).

Finally, Figure 8.11 shows the Pareto-fronts for case study 4, which is similar to case study 3, but aims to maximize the feed rate instead of minimizing SME. The Pareto curve has a different shape, since RE decreases when the output/feed rate increases, due to the reduction of the total residence time. As before, the effect of the initial screw profile is clear: RE increases with the severity of the screw profile and the Pareto curves move towards the upper right corner. The corresponding optimized screw profiles are presented in Table 8.6. In order to maximize the feed rate, it is necessary to locate all the restrictive elements towards the screw tip. This should maximize the melt temperature increase via viscous heating, which might balance the reduction in residence time resulting from increasing outputs. On the other hand, screw profiles maximizing RE yield low outputs (around 3 kg/h) and have a very different construction: the restrictive elements are generally separated by conveying elements and located near to the melting section.

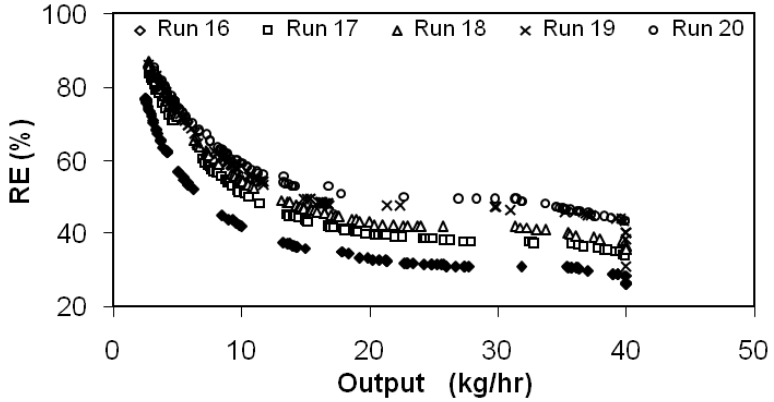

Figure 8.11. Optimal Pareto frontiers for case study 4 (runs 15 to 20, Table 7.2).

Table 8.5. Optimized screw profiles for case study 3 (*L* and *P* in mm).

Run 11: Output = 2.5 kg/hr and 37.2 kg/hr

		Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Max. RE	*L*	250 50	50	50	50	50	50	50	50	125	50	125
	P	33.3 25	KB -45°	33.3	25	33.3	16.6	16.6	25	KB -45°	16.6
Min. SME	*L*	250 50	50	50	50	50	50	125	50	50	50	125
	P	33.3 25	KB -45°	33.3	25	33.3	25	16.6	16.6	KB -45°	16.6

Run 12: Output = 2.9 kg/hr and 39.6 kg/hr

		Position	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	
Max. RE	*L*	250 50	50	50	50	50	50	50	50	50	50	50	50	50
	P	33.3 25	KB -45°	33.3	33.3	25	25	16.6	16.6	25	16.6	KB -45°	KB -45°	16.6
Min. SME	*L*	250 50	50	50	50	50	50	50	50	50	50	50	50	50
	P	33.3 25	KB -45°	33.3	33.3	25	25	16.6	25	16.6	KB -45°	16.6	KB -45°	16.6

Run 13: Output = 2.7 kg/hr and 39.8 kg/hr

		Position	1	2	3	4	5	6	7	8	9	10	11	12	13	
Max. RE	*L*	250 50	25	25	50	50	25	50	50	50	50	50	50	50	50	25
	P	33.3 25	16.6	LH 16.6	25	25	25	33.3	16.6	33.3	16.6	16.6	KB -45°	KB -45°	KB -45°	16.6
Min. SME	*L*	250 50	25	25	50	50	25	50	50	50	50	50	50	50	50	25
	P	33.3 25	16.6	LH 16.6	25	33.3	25	25	33.3	16.6	16.6	KB -45°	16.6	KB -45°	KB -45°	16.6

Run 14: Output = 2.7 kg/hr and 39.9 kg/hr

		Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Max. RE	*L*	250 50	25	25	25	50	50	25	25	50	50	25	25	50	50	50	50	25
	P	33.3 25	16.6	LH 12.5	25	33.3	25	25	25	33.3	16.6	LH 16.6	16.6	KB -45°	KB -45°	KB -45°	16.6	16.6
Min. SME	*L*	250 50	25	25	25	50	50	25	25	50	50	25	25	50	50	50	50	25
	P	33.3 25	16.6	LH 12.5	25	33.3	25	25	25	33.3	16.6	LH 16.6	16.6	KB -45°	KB -45°	KB -45°	16.6	16.6

Run 15: Output = 3.0 kg/hr and 38.9 kg/hr

		Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Max. RE	*L*	250 50	25	25	25	25	50	25	50	50	25	50	25	25	50	50	25	50	25
	P	33.3 25	16.6	LH 12.5	16.6	25	33.3	25	16.6	33.3	25	KB -45°	25	LH 16.6	KB -45°	16.6	LH 12.5	KB -45°	16.6
Min. SME	*L*	250 50	25	25	25	50	50	25	25	25	50	25	25	50	25	50	50	50	25
	P	33.3 25	16.6	LH 12.5	25	33.3	33.3	16.6	25	25	16.6	LH 16.6	25	KB -45°	LH 12.5	KB -45°	16.6	KB -45°	16.6

Table 8.6. Optimized screw profiles for case study 4 (L and P in mm).

Run 16

		Position	1	2	3	4	5	6	7	8	
Max. RE	L	250 50	50	50	50	50	50	50	125	50	125
	P	33.3 25	KB -45°	16.6	KB -45°	16.6	33.3	33.3	25	25	16.6
Min. SME	L	250 50	50	125	50	50	50	50	50	50	125
	P	33.3 25	KB -45°	25	33.3	25	16.6	33.3	16.6	KB -45°	16.6

Run 17

		Position	1	2	3	4	5	6	7	8	9	10	11	
Max. RE	L	250 50	50	50	50	50	50	50	50	50	50	50	50	50
	P	33.3 25	16.6	KB -45°	25	33.3	16.6	16.6	25	25	KB -45°	33.3	KB -45°	16.6
Min. SME	L	250 50	50	50	50	50	50	50	50	50	50	50	50	50
	P	33.3 25	KB -45°	25	16.6	25	33.3	33.3	25	16.6	16.6	KB -45°	KB -45°	16.6

Run 18

		Position	1	2	3	4	5	6	7	8	9	10	11	12	13	
Max. RE	L	250 50	25	25	50	25	50	50	50	50	50	50	50	50	50	25
	P	33.3 25	16.6	LH 16.6	33.3	25	16.6	25	25	33.3	16.6	16.6	KB -45°	KB -45°	KB -45°	16.6
Min. SME	L	250 50	25	25	50	50	50	25	50	50	50	50	50	50	50	25
	P	33.3 25	16.6	LH 16.6	33.3	25	16.6	25	33.3	16.6	25	16.6	KB -45°	KB -45°	KB -45°	16.6

Run 19

		Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Max. RE	L	250 50	25	25	50	50	25	50	50	25	50	50	50	50	25	25	25	25
	P	33.3 25	16.6	LH 12.5	KB -45°	KB -45°	25	KB -45°	16.6	LH 16.6	33.3	16.6	25	33.3	25	25	16.6	16.6
Min. SME	L	250 50	25	50	50	50	50	50	25	25	25	25	25	25	50	50	50	25
	P	33.3 25	LH 16.6	33.3	25	16.6	16.6	33.3	25	25	16.6	16.6	LH 12.5	25	KB -45°	KB -45°	KB -45°	16.6

Run 20

		Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Max. RE	L	250 50	25	25	25	25	50	50	25	25	25	50	50	25	50	50	50	25	25
	P	33.3 25	25	LH 12.5	25	LH 16.6	33.3	33.3	25	16.6	25	KB -45°	16.6	16.6	16.6	KB -45°	KB -45°	LH 12.5	16.6
Min. SME	L	250 50	25	25	25	50	50	25	50	50	25	25	25	50	25	50	50	25	25
	P	33.3 25	25	LH 12.5	25	33.3	33.3	16.6	16.6	16.6	25	16.6	25	KB -45°	LH 16.6	KB -45°	KB -45°	LH 12.5	16.6

8.5. Conclusions

The case studies discussed in this work showed the potential of multi-objective evolutionary algorithms for optimizing the screw configuration and the processing conditions for a specific application in reactive extrusion. The use of a Reduced Pareto Set Genetic Algorithm enabled the identification of feasible solutions, defining an optimal screw configuration, even when conflicting objectives were selected. The solutions proposed by the method seem to have adequate physical meaning and provide possibilities to choose the best compromise solution when opposite targets are defined.

The authors of the previous effort to optimize starch cationization demonstrated the interest of using a theoretical model for optimization, which is able to accurately describe effects that are absolutely not intuitive. The present method builds upon this previous model and can consider simultaneously various objectives, screw configuration and operating conditions, to yield optimal solutions.

Chapter 9

Conclusions and Suggestions for Further Work

9.1. Conclusions

In this thesis the definition of the optimal screw configuration of co-rotating twin screw extruders was approached as a scheduling problem. The optimization methodology applied in this study links together a modeling routine, to evaluate the quality of the solutions proposed, and an optimization algorithm, used to search for better solutions.

The global modeling routine developed considers the most important physical, thermal and rheological phenomena occurring inside an intermeshing co-rotating twin screw extruder. In order to test its accuracy and sensitivity, the computational predictions were compared with experimental data. In general, the results obtained are in agreement with theoretical and experimental knowledge and indicate that the software proposed is sensitive to variations on the operating conditions (barrel temperature, screw speed and output) and screw geometry/configuration. However, melting phenomena and flow in neutral kneading blocks need to be improved, since they were developed taking into account important simplifications.

Multi-objective local search algorithms (i.e., TPLS and PLS) are very competitive when compared with the MOEAs (namely the RPSGA) to deal with twin screw configuration problems. TPLS allow

obtaining better results for a low number of evaluations of the modeling routine. Also, PLS is much better than TPLS and RPSGA algorithms concerning for more complex problems, i.e. TSCPs problems with a high number of restrictive screw elements.

A multi-objective ant colony optimization algorithm (MOACO) was also adapted to deal with TSCPs problems. A complete study was made to define the best MOACO parameters to use in this problem. The results produced with the MOACO showed a very good performance when compared with RPSGA and TPLS algorithms.

Different combinations of the metaheuristics developed were tested in the same TSCPs. The results obtained show that the combination of MOACO and RPSGA with local search procedures (TPLS and PLS algorithms) improves considerably the quality of the multi-objective algorithms alone. Considering all the algorithms studied the hybrid MOACO+PLS is the best performing algorithm and the most appropriate to tackle the TSCP.

In all cases the solutions (screw configurations) obtained are in agreement with the current scientific and technical knowhow on the process and have physical meaning.

Finally, the multi-objective optimization methodologies proposed were tested in the optimization of a starch cationization reaction. The results obtained showed the potential of this optimization methodology in the definition of an optimal operational window (screw configuration and/or operating conditions), allowing the reduction of time and costs in practical industrial applications.

9.2. Suggestions for Further Work

Taking into consideration the results and conclusions of this work, it would be interest to pursue the following studies:

i) To perform a detailed experimental study of melting in co-rotating twin screw extruders, in order to improve the accuracy of the predictions in this plasticating step;

ii) To improve the mathematical modeling of melt flow in neutral kneading blocks;

iii) To extend the algorithms developed in this study to three and more objectives;

iv) To extend the optimization methodology to solve the TSCP when the number of elements is not known *a priori*, i.e., when a prescribed number of screw elements

must be selected from a higher available set (adjustable number of parameters to optimize) in order to occupy a certain fixed distance (length of barrel);

v) To implement several metaheuristics to deal with the problem defined in iv);

vi) To develop a methodology able to deal with a combination of discrete (location of the screw elements) and continuous parameters (individual geometry of the screw elements: thickness, staggering angle of kneading blocks, pitch and screw element length);

vii) To implement several metaheuristics, for example hybrid algorithms, to deal with the problem defined in vi).

References

ADE92 Adenso-Diaz, B., 1992. Restricted neighborhood in the tabu search for the flowshop problem. *European Journal of Operational Research,* 62 (1), 27-37.

AGA91 Agassant, J.F., Avenas, P., Sergent, J.Ph. and Carreau P., 1991. *Polymer Processing, Principles and Modeling.* Munich: Hanser Publishers.

ANG04 Angel, E., Bampis, E. and Gourvés, L., 2004. A Dynasearch Neighborhood for the Bicriteria Traveling Salesman Problem. *In:* X. Gandibleux, M. Sevaux, K. Sörensen and V. T'kindt, eds. *Metaheuristics for Multiobjective Optimisation. Lecture Notes in Computer Science,* Vol.535. Berlin: Springer Verlag.

ANG07 Angus, D., 2007. Crowding Population-based Ant Colony Optimisation for the Multi-objective Travelling Salesman Problem. *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007)*. Honolulu, 333-340.

ANG09 Angus, D. and Woodward, C., 2009. Multiple objective ant colony optimization. *Swarm Intelligence*, 3 (1), 69-85.

BAC96 Bäck, T., 1996. *Evolutionary Algorithms in Theory and Practice.* New York: Oxford University Press.

BAK87 Baker, J.E., 1987. Reducing Bias and Inefficiency in the Selection Algorithms. *In: Proceedings of the Second International Conference on Genetic Algorithms,* 14-21.

BAR03 Barán, B. and Schaerer, M., 2003. A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows. *In: Proceedings of 21st IASTED International Conference on Applied Informatics,* 97-102.

BAW95 Bawiskar, S. and White, J. L., 1995. Solids Conveying and Melting in a Starve Fed Self-Wiping Co-rotating Twin Screw Extruder. *International Polymer Processing*, 10 (2), 105-110.

BAW97 Bawiskar, S. and White, J. L., 1997. A Composite Model for Solid Conveying, Melting, Pressure and Fill factor Profiles in Modular Co-Rotating Twin Screw Extruders. *International Polymer Processing*, 12 (4), 331-340.

BAW98 Bawiskar, S. and White, J. L., 1998. Melting Model for Modular Self Wiping Co-Rotating Twin Screw Extruders. *Polymer Engineering and Science*, 38 (5), 727-740.

BAY99 Baykasõglu, A., Owen, S. and Gindy, N., 1999. A taboo search based approach to find the Pareto optimal set in multiple objective optimization. *Engineering Optimization*, 31 (6), 731-748.

BAY01 Baykasõglu, A., 2001. Goal Programming using Multiple Objective Tabu Search. *Journal of the Operational Research Society*, 52 (12), 1359-1369.

BEA93 Beasley, D., Bull, D.R. and Martin, R.R., 1993. An overview of Genetic Algorithms: Part 2, research Topics. *University Computing*, 15 (4), 170-181.

BEN92 Bentley, J.L., 1992. Fast Algorithms for Geometric Traveling Salesman Problems. *ORSA Journal on Computing*, 4 (4), 387–411.

BER98 Berzin, F. and Vergnes, B., 1998. Transesterification of ethylene acetate copolymer in a twin screw extruder. *International Polymer Processing*, 13 (1), 13-22.

BER00 Berzin, F., Vergnes, B., Dufossé, P., and Delamare, L., 2000. Modelling of peroxide initiated controlled degradation of polypropylene in a twin screw extruder. *Polymer Engineering and Science*, 40, 344-356.

BER06 Berzin, F., Vergnes, B., Canevarolo, S.V., Machado, A.V., and Covas, J.A., 2006. Evolution of peroxide-induced degradation of polypropylene along a twin-screw extruder: experimental data and theoretical predictions. *Journal of Applied Polymer Science*, 99 (5), 2082-2090.

BER07a Berzin, F., Tara, A., Tighzert, L. and Vergnes, B., 2007. Computation of starch cationization performances by twin-screw extrusion. *Polymer Engineering and Science*, 47 (2), 112-119.

BER07b Berzin, F., Tara, A. and Vergnes, B., 2007. Optimization and scale-up of starch cationization in a twin screw extruder. *Polymer Engineering and Science*, 47 (6), 814-823.

BER07c Berzin, F., Tara, A. and Tighzert, L., 2007. In-line measurement of the viscous behaviour of wheat starch during extrusion. Application to starch cationisation. *Applied Rheology*, 17, 21222-21229

BLI95 Blickle, T. and Thiele, L., 1995. A Comparison of Selection Schemes Used in Genetic Algorithms. TIK – Report N° 11, Computer Engineering and Communications Network Lab, Zurich, Switzerland.

BLU03 Blum, C. and Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35 (3), 268–308.

BLU05 Blum, C., Roli, A., Alba, E., 2005. An introduction to metaheuristic techniques. *In:* E. Alba, ed. *Parallel Metaheuristics, a New Class of Algorithms*. New Jersey: John Wiley, 3–42.

BLU08 Blum, C., Blesa Aguilera, M.J., Roli, A. and Sampels, M. (eds.), 2008. *Hybrid Metaheuristics An Emerging Approach to Optimization. Studies in Computational Intelligence,* Vol.114. Berlin: Springer-Verlag.

BLU10 Blum, C., Puchinger, J., Roli, A., Raidl, G.R., 2010. A Brief Survey on Hybrid Metaheuristics. *Informatica*, 3-16.

BOO78 Booy, M. L., 1978. Geometry of Fully Wiped Twin-Screw Equipment. *Polymer Engineering and Science*, 18 (12), 973-984.

BOO80 Booy, M. L., 1980. Isothermal Flow of Viscous Liquids in Corotating Twin Screw Devices. *Polymer Engineering and Science*, 20 (18), 1220-1228.

BRA00 Bravo, V.L., Hrymak, A.N. and Wright, J.D., 2000. Numerical Simulation of Pressure and Velocity Profiles in Kneading Elements of a Co-Rotating Twin Screw Extruder. *Polymer Engineering and Science*, 40 (2), 525-541.

CAN99 Canedo, E.L., 1999. Computer Simulation of Plastics Compounding Operations in Twin-Screw Extruders. *SPE Antec Technical Papers*, 45, 310-316.

CAR03 Cardoso, P., Jesus, M. and Marquez, A., 2003. MONACO - Multi-Objective Network Optimisation based on ACO. *X Encuentros de Geometría Computacional.* Seville, Spain.

CAR99 Carneiro, O.S., Caldeira, G. and Covas, J.A., 1999. Flow patterns in twin screw extruders. *Journal of Materials Processing Technology*, 92-93, 309-315.

CAR00 Carneiro, O.S., Covas, J.A. and Vergnes, B., 2000. Experimental and Theoretical Study of Twin Screw Extrusion of Polypropylene. *Journal of Applied Polymer Science*, 78, 1419-1430.

CAR02 Carneiro, O.S., Poulesquen, A., Covas, J.A. and Vergnes, B., 2002. Visualization and Analysis of the Flow along the Kneading Block of a Twin-screw Extruder. *International Polymer Processing*, 17 (4), 301-308.

CAR04 Carneiro, O.S.; Covas, J.A.; Ferreira, J.A. and Cerqueira, M.F., 2004. On-line monitoring of the residence time distribution along a kneading block of a twin-screw extruder. *Polymer Testing*, 23 (8), 925-937.

CAR93 Carrot, C., Guillet, J., May, J.F. and Puaux, J.P., 1993. Modeling of the Conveying of Solid Polymer in the Feeding Zone of Intermeshing Co-rotating Twin Screw Extruders. *Polymer Engineering and Science*, 33 (11), 700-708.

CER85 Cerný, V., 1985. A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45 (1), 41-51.

CHE94a Chen, Z. and White, J.L., 1994. Simulation of Non-Isothermal Flow in Modular Co-Rotating Twin Screw Extrusion. *Polymer Engineering and Science*, 34 (3), 229-237.

CHE94b Chen, Z. and White, J.L., 1994. Simulation of Non-Isothermal Flow in Twin Screw Extrusion. *International Polymer Processing*, 9 (4), 310-318.

CHE97 Cheng, H. and Manas-Zloczower, I., 1997. Study of Mixing Efficiency in Kneading Discs of Co-Rotating Twin-Screw Extruders. *Polymer Engineering and Science*, 37 (6), 1082-1090.

CHE98 Cheng, H. and Manas-Zloczower, I., 1998. Distributive Mixing in Conveying Elements of a ZSK-53 Co-rotating Twin Screw Extruder. *Polymer Engineering and Science*, 38(6), 926-935.

CHI96 Chiruvella, R.V., Jaluria, Y., Karwe, M.V. and Sernas, V., 1996. Transport in a Twin Screw Extruder For The Processing of Polymers. *Polymer Engineering and Science*, 36 (11), 1531-1540.

CHO04 Choulak, S., Couenne, F., Le Gorrec, Y., Jallut, C., Cassagnau, P. and Michel, A., 2004. Generic Dynamic Model for Simulation and Control of Reactive Extrusion. *Industrial & Engineering Chemical Research,* 43 (23), 7373–7382.

CHU98 Chung, C., 1998. A Scientific Approach to Screw Design. *Journal of Reinforced Plastics and Composites,* 17 (12), 1096-1109.

COE07 Coello Coello, C.A., Lamont, G: and Van Veldhuizen, D.A., 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Springer.

CON99 Conover, W. J., 1999. *Practical Nonparametric Statistics*, 3rd edn. New York: John Wiley & Sons.

COT98 Cotta, C., 1998. A Study of Hybridisation Techniques and their Application to the Design of Evolutionary Algorithms. *AI Communications*, 11 (3-4), 223-224.

COV95 Covas, J.A., 1995. The Inverse Problem in Polymer Processing. *In:* J.A. Covas, J.F. Agassant, A.C. Diogo, J. Vlachopoulos and K. Walters, eds. *Rheological fundamentals of polymer processing.* Dordrecht: Kluwer Academic Publishers, 385-406.

COV99 Covas, J.A., Gaspar-Cunha, A. and Oliveira, P., 1999. An Optimization Approach to Practical Problems in Plasticating Single Screw Extrusion. *Polymer Engineering and Science,* 39 (3), 443-456.

CZY97 Czyzak, P. and Jaszkiewicz, A., 1997. Pareto Simulated Annealing. *In*: G. Fandel and T. Gal, eds. *Multiple Criteria Decision Making. Proceedings of the XIIth International Conference*. Germany: Springer- Verlag, 297–307.

CZY98 Czyzak, P. and Jaszkiewicz, A., 1998. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7, 34–47.

DAR59 Darwin, C., 1859. *On the Origin of Species by Means of Natural Selection.*

DEB89 Deb, K. and Goldberg, D.E., 1989. An Investigation of Niche and Species Formation in Genetic Function Optimization, *In: Proceedings of the Third International Conference on Genetic Algorithms*. San Francisco, CA: Morgan Kauffman, 41-49.

DEB00 Deb, K., Agrawal, S., Meyarivan, T. and Pratap, A., 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGAII. *In*: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo and H.-P. Schwefel, eds. *Parallel Problem Solving from Nature (PPSN VI). Lecture Notes in Computer Science,* Vol.1917. Heidelberg, Germany: Springer, 849 – 858.

DEB01 Deb K., 2001. *Multi-objective Optimization using Evolutionary Algorithms*. Chichester: Wiley.

DEB02 Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., 2002. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6 (2), 182-197.

DEL93 Della Valle, G., Barres, C., Plewa, J., Tayeb, J. and Vergnes, B., 1993. Computer simulation of starchy products' transformation by twin screw extrusion. *Journal of Food Engineering*, 19 (1), 1-31.

DEN80 Denson, C.D. and Hwang, B.K., 1980. The Influence of the Axial Pressure Gradient on flow rate for Newtonian Liquids in a Self-Wiping Co-Rotating Twin Screw Extruder. *Polymer Engineering and Science*, 20 (14), 965-971.

DOE03 Doerner, K., Hartl, R.F. and Reimann, M., 2003. Are COMPETants more competent for problem solving? – The Case of Full Truckload Transportation. *Central European Journal of Operations Research,* 11 (2), 115–141.

DOE04 Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C. and Stummer, C., 2004. Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. *Annals of Operations Research*, 131 (3), 79-99.

DOM10 Domingues, N., Gaspar-Cunha, A. and Covas J.A., 2010. Estimation of the Morphology Development of Immiscible Liquid–liquid Systems during Single Screw Extrusion. *Polymer Engineering and Science*, 50 (11), 2194-2204.

DOR92 Dorigo, M. 1992. *Optimization, Learning and Natural Algorithms.* PhD thesis, Politecnico di Milano, Italy.

DOR97 Dorigo, M. and Gambardella, L.M., 1997. Ant Colony System: A cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1 (1), 53-66.

DOR99 Dorigo, M. and Di Caro, G., 1999. The Ant Colony Optimization Metaheuristic. *In:* D. Corne, M. Dorigo and F. Glover, eds. *New Ideas in Optimization*. London, UK: McGraw-Hill, 1-32.

DOR04 Dorigo, M. and Stützle, T., 2004. *Ant Colony Optimization.* Cambridge: MIT Press.

DOR10 Dorigo, M. and Stützle, T., 2010. Ant Colony Optimization: Overview and recent advances. *In:* M. Gendreau and Y. Potvin, eds. *Handbook of Metaheuristics. International Series in Operations Research & Management Science,* 2nd ed., Vol.146. New York: Springer, 227-263.

DUB10 Dubois-Lacoste, J., López-Ibáñez, M. and Stützle, T., 2010. Adaptive ''Anytime'' Two-Phase Local Search. *In:* C. Blum and R. Battiti, eds. *Learning and Intelligent Optimization, Fourth International Conference (LION 4). Lecture Notes in Computer Science*, *Vol.6073.* Heidelberg: Springer, 52–67.

DUB12 Dubois-Lacoste, J., López-Ibáñez, M. and Stützle, T., 2012. Pareto Local Search Algorithms for Anytime Bi-Objective Optimization. *In:* J-K Hao and M. Middendorf, eds. *Proceedings of EvoCOP 2012 – 12th European Conference on Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science,* Vol.7245. Heidelberg: Springer, 206-217.

DUB13 Dubois-Lacoste, J., López-Ibáñez, M. and Stützle, T., 2013. Combining Two Search Paradigms for Multi-objective Optimization: Two-Phase and Pareto Local Search. *In*: E. Talbi, ed. *Hybrid Metaheuristics, Studies in Computational Intelligence*, Vol. 434, 97-117.

EAS23 Easton, R.W., 1923. US Patent 1468379.

EDG81 Edgeworth, F.Y., 1881*. Mathematical Physics*.

EHR08 Ehrgott, M., Gandibleux, X., 2008. Hybrid Metaheuristics for Multi-objective Combinatorial Optimization. *In*:. C. Blum, M. Aguilera, A. Roli and M. Sampels, eds. *Hybrid Metaheuristics*, 114, 221-259.

ELB84 Elbirli, B., Lindt, J.T., Gottgetreu, S. R. and Baba, S. M.*,* 1984. Mathematical Modelling of Melting of Polymers in Single-Screw Extruder. *Polymer Engineering and Science,* 24 (12), 988-999.

ERD64a Erdmenger, R., 1964. US Patent 3122356.

ERD64b Erdmenger, R., 1964. Mehrwellen-Schnecken in der Verfahrenstechnik. *Chemie Ingenieur Technik*, 36 (3), 175-185.

FAR10 Fard, A.S., 2010. Analysis and optimization of mixing in twin screw extruders, PhD Thesis, Eindhoven University of Technology.

FER10 Fernandes, C., Pontes, A., Viana, J. and Gaspar-Cunha, A., 2010. Using multiobjective evolutionary algorithms in the optimization of operating conditions of polymer injection molding. *Polymer Engineering and Science*, 50 (8), 1667-1678.

FER11 Fernandes, C., Pontes, A., Viana, J. and Gaspar-Cunha, A., 2011. Using Multi-Objective Evolutionary Algorithms for Optimization of the Cooling System in Polymer Injection Moulding. *International Polymer Processing*, 1, 1-10.

FOG62 Fogel, L.J., 1962. Toward inductive inference automata. *In: Proceedings of the International Federation for Information Processing Congress.* Munich, 395-399.

FOG66 Fogel, L.J., Owens, A.J. and Walsh, M.J., 1966. *Artificial Intelligence through Simulated Evolution,* Whiley.

FON93 Fonseca, C.M. and Fleming, P., 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *In:* S. Forrest, ed. *Genetic Algorithms: Proceedings of the Fifth International Conference.* San Mateo, California: Morgan Kaufmann, 416-423.

FON96 Fonseca, C.M. and Fleming, P., 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. *In:* H.M. Voigt, W. Ebeling, I. Rechenberg and H.P. Schwebel, eds. *Proceedings of Fourth International Conference on Parallel Problem Solving from Nature (PPSN-IV). Lecture Notes in Computer Science*, Vol.1141. Heidelberg: Springer, 584–593.

FUK00 Fukuoka, T., 2000. Numerical analysis of a reactive extrusion process. Part II: Simulations and verifications for the twin screw extrusion. *Polymer Engineering and Science*, 40 (12), 2524-2538.

GAN97 Gandibleux, X., Mezdaoui, N. and Fréville, A., 1997. A Tabu Search Procedure to Solve Combinatorial Optimisation Problems. *In:* R. Caballero, F. Ruiz, and R. E. Steuer, eds. *Advances in Multiple Objective and Goal Programming. Lecture Notes in Economics and Mathematical Systems,* Vol. 455. Spain: Springer-Verlag, 291-300.

GAR07 García-Martínez, C., Córdon, O. and Herrera, F., 2007. A Taxonomy and an Empirical Analysis of Multiple Objective Ant Colony Optimization Algorithms for the Bi-Criteria TSP. *European Journal of Operational Research*, 180 (1), 118-148.

GAS97 Gaspar-Cunha, Oliveira, P. and Covas, J.A., 1997. Use of Genetic Algorithms in Multicriteria Optimization to Solve Industrial Problems. *In:* T. Bäck, ed. *Proceedings of Seventh International Conference on Genetic Algorithms (ICGA97)*. Morgan Kaufmann Publishers, 682-688.

GAS00 Gaspar-Cunha, A., 2000. *Modelling and Optimization of Single Screw Extrusion*. PhD Thesis. University of Minho, Portugal.

GAS01 Gaspar-Cunha, A., Vergnes, B. and Covas, J.A., 2001. The Design of Extrusion Screws: An Optimization Approach. *International Polymer Processing,* 16 (3), 229-240.

GAS02 Gaspar-Cunha, A., Poulesquen, A., Vergnes B. and Covas, J. A., 2002. Optimization of Processing Conditions for Polymer Twin-Screw Extrusion. *International Polymer Processing,* 17 (3), 201-213.

GAS04 Gaspar-Cunha, A.; Covas, J.A.. 2004. RPSGAe–Reduced Pareto Set Genetic Algorithm: Application to Polymer Extrusion. *In:* X. Gandibleux, M. Sevaux, K. Sörensen and V. T'kindt, eds. *Metaheuristics for Multiobjective Optimisation. Lectures Notes in Economics and Mathematical Systems,* Vol.535. Heidelberg: Springer, 221-249.

GAS05 Gaspar-Cunha, A., Vergnes, B. and Covas, J. A., 2005. Defining the Configuration of Co-Rotating Twin-Screw Extruders with Multi-Objective Evolutionary Algorithms. *Polymer Engineering and Science*, 45 (8), 1159-1173.

GAS06 Gaspar-Cunha, A., Gonçalves, L. and Covas, J.A., 2006. Application of Evolutionary Algorithms to the Design of Barrier Screws for Single Screw Extruders. *In:* A. Abraham, B. de Baets, M. Köppen and B. Nickolay, eds. *Applied Soft Computing Technologies: The Challenge of Complexity*. Advances in Soft Computing. Heidelberg: Springer, 763-774.

GAS11 Gaspar-Cunha, A., Covas, J.A., Vergnes, B. and Berzin, F., 2011. Reactive Extrusion - Optimization of Representative Processes, *In:* A. Gaspar-Cunha and J.A. Covas, eds. *Optimization in Polymer Processing*. USA: Nova Science Publishers.

GLO86 Glover, F, 1986. Future Paths for Integer Programming and links to Artificial Intelligence. *Computers and Operations Research,* 13 (5), 553-549.

GLO89 Glover, F., 1989. Tabu Search - Part I. *ORSA Journal on Computing,* 1 (3), 190-206.

GLO90 Glover, F., 1990. Tabu Search - Part II. *ORSA Journal on Computing*, 2 (1), 4-32.

GLO95 Glover, F., Kelly, J.P. and Laguna, M., 1995. Genetic Algorithms and Tabu Search: Hybrids for Optimization. *Computers and Operations Researcher*, 22 (1), 111-134.

GLO97 Glover, F. and Laguna, M., 1997. *Tabu Search*. Norwell, MA: Kluwer Academic Publishers.

GOF96 Goffart, D., Van der Wal, D.J., Klomp, E.M., Hoogstraten, H.W., Janssen, L.P.B.M., Breysse, L. and Trolez, Y., 1996. Three-dimensional Flow Modeling Of A Self-Wiping Corotating Twin-Screw Extruder. Part 1: The Transportation Section. *Polymer Engineering and Science,* 36 (7), 901-911.

GOG98 Gogos, C.G., Tadmor, Z. and Kim, M.H., 1998. Melting Phenomena and Mechanisms in Polymer Processing Equipment. *Advances in Polymer Technology*, 17 (4), 285-305.

GOL89 Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Boston, MA: Addison-Wesley.

GOR91 Gorges-Schleuter, M., 1991. Explicit parallelism of Genetic Algorithms through Population Structures. *In:* H.P. Schwefel and R. Männer, eds. *Parallel Problem Solving from Nature. Lectures Notes in Computer Science,* Vol.496. Berlin: Springer-Verlag, 150-159.

GOT90 Gotsis, A.D., Ji, Z. and Kalyon, D.M., 1990. Numerical Simulation of the Flow in the Kneading Block Region of the Co-rotating Twin Screw Extruder. *SPE Antec Technical Papers*, 36, 139.

GRU01 Grunert da Fonseca, V., Fonseca, C.M. and Hall, A., 2001. Inferential performance assessment of stochastic optimizers and the attainment function. *In:* E. Zitzler, K. Deb, L. Thiele, C.C. Coello and D. Corne, eds. *Evolutionary Multicriterion Optimization (EMO 2001). Lecture Notes in Computer Science,* Vol.1993. Heidelberg, Germany: Springer, 213–225.

GRU10 Grunert da Fonseca, V. and Fonseca, C.M., 2010. The attainment-function approach to stochastic multiobjective optimizer assessment and comparison. *In:* T. Bartz-Beielstein, M. Chiarandini, L. Paquete and M. Preuss, eds. *Experimental Methods for the Analysis of Optimization Algorithms*. Heidelberg: Springer, 103-130.

GUN03 Guntsch, M. and Middendorf, M., 2003. Solving Multi-criteria Optimization Problems with Population-Based ACO. *In:* C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, eds. *Proceedings of the 2nd International Conference on Evolutionary Multi-criterion Optimization (EMO 2003). Lecture Notes in Computer Science*, Vol. 2632. Heidelberg: Springer, 464-478.

HAN97 Hansen, M.P., 1997. Tabu Search in Multiobjective Optimisation: MOTS. *In: Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM'97)*. Heidelberg: Springer, 6-10.

HAN00 Hansen, M., 2000. Tabu search for multiobjective combinatorial optimization: TAMOCO. *Control and Cybernetics*, 29 (3), 799–818.

HER94 Hertz, A., Jaumard, B., Ribeiro, C. and Filho, W.F., 1994. A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives. *RAIRO/Operations Research*, 28 (3), 303–328.

HO02 Ho, S., Yang, S., Ni, G. and Wong, H., 2002. A Tabu Method to Find the Pareto Solutions of Multiobjective Optimal Design Problems in Electromagnetics. *IEEE Transactions on Magnetics*, 38 (2), 1013–1016.

HOL75 Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems.* Ann Harbor, MI: University of Michigan Press.

HOO04 Hoos, H. and Stützle, T., 2004. *Stochastic Local Search – Foundations and Applications*. San Francisco, CA: Morgan Kaufmann Publishers.

IRE01 Iredi, S., Merkle, D. and Middendorf, M., 2001. BiCriterion Optimization with Multi Colony Ant Algorithms. *In:* E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello and D. Corne, eds. *First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science*, Vol. 1993. Heidelberg: Springer-Verlag, 358-372.

ISH00 Ishikiwa, T., Nagashima, Y. and Funatsu, K., 2000. 3-D Numerical Simulations of Nonisothermal Flow in Co-rotating Twin Screw Extruders, *Polymer Engineering and Science*, 40 (2), 357-364.

JAE04 Jaeggi, D., Asselin-Miller, C., Parks, G., Kipouros, T., Bell, T. and Clarkson, J., 2004. Multi-objective Parallel Tabu Search. *In:* X. Yao, E.K. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J.E. Rowe, P. Tino, A. Kabán and H-P. Schwefel, eds. *Parallel Problem Solving from Nature - PPSN VIII. Lecture Notes in Computer Science*, Vol.3242. Heidelberg: Springer-Verlag, 732-741.

JUN03 Jung, H. and White, J.L., 2003. Investigation of Melting Penomena in Modular Co-Rotating Twin Screw Extrusion, *International Polymer Processing*, 18 (2), 127-132.

JUN08 Jung, H. and White, J.L., 2008. Modeling and Simulation of the Mechanisms of Melting in a Modular Co-Rotating Twin Screw Extruder. *International Polymer Processing*, 23 (3), 242-251.

KAC72 Kacir, L. and Tadmor, Z., 1972. Solids Conveying in Screw Extruders – Part III: The Delay Zone. *Polymer Engineering and Science*, 12 (5), 387-395.

KIM00 Kim, E.K. and White, J.L., 2000. Non-isothermal Transient Startup of A Starved Flow Modular Co-Rotating Twin Screw Extruder, *International Polymer Processing*, 15(3), 233-241.

KIR83 Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P., 1983. Optimization by simulated annealing. *Science*, 220 (4598), 671—680.

KNO99 Knowles, J. and Corne, D., 1999. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99).* Washington: IEEE ServiceCenter, 98-105.

KNO00 Knowles, J. and Corne, D., 2000. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8 (2), 149-172.

KNO06 Knowles, J., Thiele, L. and Zitzler, E., 2006. A tutorial on the performance assessment of stochastive multiobjective optimizers. Technical Report TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich.

KOH07 Kohlgrüber, K., 2007. *Co-rotating twin-screw extruders: Fundamentals, Technology, and Applications*. Munich, Germany: Hanser.

KOZ91 Koza, J.R., 1991. Evolving a computer Program to Generate Random numbers Using the Genetic Programming Paradigm. *Proceedings of the Fourth International Conference on Genetic Algorthms.* Morgan Kauffman, 37-44.

KUL06 Kulturel-Konak, S., Smith, A. E., and Norman, B. A., 2006. Multi-objective tabu search using a multinomial probability mass function. *European Journal of Operational Research*, 169 (3), 918–931.

KYE96 Kye, H. and White, J. L., 1996. Simulation of Continuous Polymerization in a Modular Intermeshing Co-rotating Twin Screw Extruder. *International Polymer Processing*, 11 (2), 129-138.

LAU02 Laumanns, M., Thiele, L., Zitzler, E., Welzl, E. and Deb, K., 2002. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. *In:* J. Guervós, P. Adamidis, H.-G. Beyer, J. Martins and H.-P. Schwefer, eds. *Parallel Problem Solving From Nature (PPSN IV). Lecture Notes of Computer Science,* Vol. 2439. Berlin: Springer, 44-53.

LIU01 Liu, T., Wong, A.C.–Y. and Zhu, F., 2001. Prediction of Screw Length Required for Polymer Melting and Melting Characteristics. *International Polymer Processing*, 16 (2), 113-123.

LOP06 López-Ibáñez, M., Paquete, L. and Stützle, T., 2006. Hybrid population-based algorithms for the bi-objective quadratic assignment problem. *Journal of Mathematical Modelling and Algorithms,* 5 (1), 111–137.

LOP09 López-Ibáñez, M. and Stützle, T., 2009. An analysis of algorithmic components for multiobjective ant colony optimization: A case study on the biobjective TSP. *In:* P. Collet, N. Monmarché, P. Legrand, M. Schoenauer and E. Luttoon, eds. *Artificial Evolution-9th International Conference, Evolution Artificielle (EA'2009)*. *Lecture Notes in Computer Science*, Vol.5975. Heidelberg, Germany: Springer, 134-145.

LOP10a López-Ibáñez, M., Paquete, L. and Stützle, T., 2010. Exploratory analysis of stochastic local search algorithms in biobjective optimization. *In:* T. Bartz-Beielstein, M. Chiarandini, L. Paquete and M. Preuß, eds. *Experimental Methods for the Analysis of Optimization Algorithms*. Berlin, Germany: Verlag, 209–233.

LOP10b López-Ibáñez, M. and Stützle, T., 2010. The Impact of Design Choices of Multiobjective Ant Colony Optimization Algorithms on Performance: An Experimental Study on the Biobjective TSP. *In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010).* New York, NY: ACM Press, 71-78.

LOU02 Lourenço, H. R, Martin, O. and Stützle, T., 2002. Iterated local search. *In:* F. Glover and G. Kochenberger, eds. *Handbook of Metaheuristics*. *International Series in Operation Research and Management Science,* Vol.57. Norwell, MA: Kluwar Academic Publishers, 321-353.

MAC99 Machado, A. V., Covas, J. A. and Duin, M. van, 1999. Evolution of Morphology and of Chemical Conversion along the Screw in a Corotating Twin Screw Extruder. *Journal of Applied Polymer Science*, 71 (1), 135-141.

MAD59 Maddock, B.H., 1959. A Visual Analysis of Flow and Mixing in Extruder Screws. *SPE Journal*, 15, 383-389.

MAR99a Mariano, C.E. and Morales, E., 1999. MOAQ an Ant-Q Algorithm for Multiple Objective Optimization Problems. *In:* W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela and R. E. Smith, eds. *Genetic and Evolutionary Computing Conference (GECCO99)*, Vol.1. San Francisco, California: Morgan Kaufmann, 894-901.

MAR99b Mariano, C. and Morales, E., 1999. *A Multiple Objective Ant-Q Algorithm for the Design of Water Distribution Irrigation Networks*. Technical Report HC-9904, Instituto Mexicano de Tecnologia Del Agua.

MEI88 Meijer, H. E. H. and Elemans, P. H. M., 1988. The Modeling of Continuous Mixers. Part I: The Corotating Twin Screw Extruder. *Polymer Engineering and Science*, 28 (5), 275-290.

MER00 Merkle, D. and Middendorf, M., 2000. An Ant Colony with a New Pheromone Evaluation Rule for Tardiness Problems. In: S. Cagnoni, R. Poli, G. D. Smith, D. Cornes, M. Oates, E. Hart, P. L. Lanzi, E. J. Willem, Y. Li, B. Paechter and T. C. Fogarty, eds. *Proceedings of the Evo Workshops 2000*. *Lecture Notes in Computer Science,* Vol.1803. Heidelberg, Germany: Springer. 287-296.

MET53 Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E., 1953. Equation of State Calculation by fast Computing Machines. *Journal of Chemical Physics*, 21: 1087-1092

MIE99 Miettinen, K., 1999. *Nonlinear Multiobjective Optimization*, Boston: Kluwer.

MOS99 Moscato, P., 1999. Memetic Algorithms: A Short Introduction. *In:* D. Corne and M. Dorigo and F. Glover, eds. *New Ideas in Optimization*, 219-234.

MUH88 Mühlenbein, H., Gorges-Schleuter, M. and Kräme, O., 1988. Evolution Algorithms in Combinatorial Optimization. *Parallel Computing,* 7 (1), 65-85.

PAR96 Pareto, V., 1896. *Cours D´Economie Politique*.

PAP82 Papadimitriou, C. H. and Steiglitz, K., 1982. *Combinatorial Optimization – Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice Hall.

PAQ03 Paquete, L. and Stützle, T., 2003. A Two-Phase Local Search for the Biobjective Traveling Salesman Problem. *In:* C. Fonseca, P., Fleming, E., Zitzler, K., Deb and L. Thiele, eds. *Evolutionary Multi-criterion Optimization (EMO 2003). Lecture Notes in Computer Science*, Vol.2632. Heidelberg: Springer Verlag, 479–493.

PAQ04 Paquete, L., Chiarandini, M. and Stützle, T., 2004. Pareto Local Optimum Sets in the Biobjective Traveling Salesman Problem: An Experimental Study. *In:* X. Gandibleux, M., Sevaux, K., Sorensen and V. T'Kindt, eds. *Metaheuristics for Multiobjective Optimisation*. *Lecture Notes in Economics and Mathematical Systems,* Vol.535. Heidelberg: Springer Verlag, 177-199.

PAQ05 Paquete, L. and Stützle, T., 2005. A study of local search algorithms for the biobjective QAP with correlated flow matrices. *European Journal of Operational Research,* 169 (3), 943–959.

PAQ07 Paquete, L., Schiavinotto, T. and Stützle, T., 2007. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research*, 156 (1), 83–98.

POT94 Potente, H., Ansahl, J. and Klarholz, B., 1994. Design of Tightly Intermeshing Co-Rotating Twin Screw Extruders. *International Polymer Processing*, 9 (1), 11-25.

POT96a Potente, H. and Melisch, U., 1996. Theoretical and Experimental Investigations of the Melting of Pellets in Co-Rotating Twin Screw Extruders. *International Polymer Processing*, 11 (2), 101-108.

POT96b Potente, H., Melisch, U. and Palluch, K.P., 1996. A Physico-Mathematical Model for Solids Conveying in Co-Rotating Twin Screw Extruders. *International Polymer Processing*, 11 (1), 29-41.

POT99 Potente, H., Bastian, M. and Flecke, J., 1999. Design of a Compounding Extruder by means of the SIGMA Simulation Software. *Advances in Polymer Technology*, 18 (2), 147-170.

POT06 Potente, H. and Thumen, A., 2006. Method for the optimization of screw elements for tightly intermeshing, co-rotating twin screw extruders. *International Polymer Processing*, 21 (2) 149-154.

POU01a Poulesquen, A., 2001. *Contribution à la modélisation de l'extrusion réactive en machine bi-vis. PhD dissertation, Ecole des Mines de Paris*, Sophia-Antipolis, France.

POU01b Poulesquen, A., Vergnes, B., Cassagnau, P., Gimenez, J. and Michel, A., 2001. Polymerization of ε-caprolactone in a twin screw extruder: experimental study and modeling. *International Polymer Processing*, 16 (1), 31-38.

PUC05 Puchinger, J. and Raidl, G. R., 2005. Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. *In:* J. Mira and J. R. Álvarez, eds. *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach. Lecture Notes in Computer Science,* Vol. 3562, 41-53.

QIA00 Qian, B. and Gogos C. G., 2000. The Importance of Plastic Energy Dissipation (PED) to the Heating and Melting of Polymer Particulates in Intermeshing Co-Rotating Twin Screw Extruders. *Advances in Polymer Technology*, 19 (4), 287-299.

QIA03 Qian, B., Todd, D. B. and Gogos, C. G., 2003. Plastic Energy Dissipation and Its Role on Heating/Melting of Single-Component Polymers and Multi-Component Polymer Blends. *Advances in Polymer Technology*, 22 (2), 85-95.

RAI06 Raidl, G.R., 2006. A Unified View on Hybrid Metaheuristics. *In:* F. Almeida, M. J. Blesa Aguilera, C. Blum, J. Moreno Vega, M. Pérez Pérez, A. Roli and M. Sampels, eds. *Hybrid Metaheuristics. Lectures Notes in Computer Science,* Vol.4030, 1-112.

RAU86 Rauwendaal, C., 1986. *Polymer Extrusion*. Munich: Hanser Publishers.

REC73 Rechenberg, I., 1973. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Struttgart: Fromman-Holzboog.

SAK91 Sakai, T., 1991. Report on the state of the art: Reactive processing using twin-screw extruders*. Advances in Polymer Technology*, 11 (2), 99-108

SAK95 Sakai, T., 1995. The development of on-line techniques and novel processing systems for the monitoring and handling of the evolution of microstructure in nonreactive and reactive polymer systems. *Advances in Polymer Technology*, 14 (4), 277-290.

SCH84 Schaffer, J.D., 1984. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. PhD thesis. Vanderbilt University, Nashville, Tennessee.

SER94 Serafini, P., 1994. Simulated Annealing for Multiple Objective Optimization Problems. *In*: G. Tzeng, H. Wang, U. Wen and P. Yu, eds. *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application*, Vol.1. Berlin: Springer-Verlag, 283-292.

SRI94 Srinivas, N. and Deb, K., 1994. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms.*Evolutionary Computation*, 2 (3), 221–248.

STR00 Strutt, D., Tzoganakis, C. and Duever, T.A., 2000. Mixing Analysis of Reactive Polymer Flow in Conveying Elements of a Co-Rotating Twin-Screw Extruder. *Advances in Polymer Technology*, 19 (1), 22-33.

STU98a Stützle, T., 1998. *Local Search Algorithms for Combinatorial Problems – Analysis, Improvements, and New Applications*. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany.

STU98b Stützle, T. and Hoos, H., 1998. Improvements on the Ant System: Introducing the MAX-MIN Ant System. *In:* G.D. Smith, N. C. Steele and R. F. Albrecht, eds. *Proceedings of Artificial Neural Nets and Genetic Algorithms*. Vienna, Austria: Springer, 245-249.

STU00 Stützle, T. and Hoos, H. 2000. MAX-MIN Ant System. *Future Generation Computer Systems*, 16 (8), 889-914.

SUM02 Suman, B., 2002. Multiobjective simulated annealing – A metaheuristic technique for multiobjective optimization of a constrained problem. *Foundations of Computing and Decision Sciences*, 27 (3), 171–191.

SUP00 Suppapitnarm, A., Seffen, K., Parks, G. and Clarkson, P., 2000. A simulated annealing algorithm for multiobjective optimization. *Engineering Optimization*, 33 (1), 59–85.

SZY87a Szydlowski, W. and White, J. L., 1987. An Improved Theory of Metering in an Intermeshing Corotating Twin-Screw Extruder. *Advances in Polymer Technology*, 7 (2), 177-183.

SZY87b Szydlowski, W., Brzoskowski, R. and White, J. L., 1987. Modelling Flow in an Intermeshing Co-Rotating Twin Screw Extruder: Flow in Kneading Discs. *International Polymer Processing*, 1 (4), 207-214.

SZY88a Szydlowski, W. and White, J. L., 1988. Improved Model of Flow in the Kneading Disc Region of a Modular Intermeshing Co-Rotating Twin Screw Extruder. *International Polymer Processing*, 2 (3/4), 142-150.

SZY88b Szydlowski, W. and White, J.L., 1988. A Non-Newtonian Model of Flow in a Kneading Disc Region of a Modular Intermeshing Corotating Twin Screw Extruder. *Journal of Non-Newtonian Fluid Mechanics*, 28 (1), 29-46.

TAD70 Tadmor, Z. and Klein, I., 1970. *Engineering Principles of Plasticating Extrusion*. New York: Nostrand Reinhold.

TAD72 Tadmor, Z. and Broyer, 1972. Solids Conveying in Screw Extruders – Part II: Non Isothermal Model. *Polymer Engineering Science*, 12 (5), 378-386.

TAD74 Tadmor, Z., Broyer, E. and Gulfinger, C., 1974. Flow Analysis Network (FAN) – A Method for Solving Flow Problems in Polymer Processing. *Polymer Engineering and Science*, 14 (9), 660-665.

TAD06 Tadmor, Z. and Gogos, C. G.., 2006. *Principles of Polymer Processing*. New York: John Wiley & Sons.

TAL02 Talbi, E. G., 2002. A taxonomy of hybrid metaheuristics. *Journal of Heuristics,* 8 (5), 541-565.

TAO98 Tao, G. and Michalewicz, Z., 1998. Inver-over Operator for the TSP. *In:* T. Baeck, A. E. Eiben, M. Schoenauer and H.-P. Schwefel, eds. *Parallel Problem Solving from Nature (PPSN V). Lecture Notes in Computer Science*, Vol.1498. Heidelberg, Germany: Springer, 803-812.

TAR04 Tara, A., Berzin, F., Tighzert, L. and Vergnes, B., 2004. Preparation of Cationic Wheat Starch by Twin-Screw Reactive Extrusion. *Journal. of Applied Polymer Science*, 93, 201-208.

TEG03 Teghem, J. and Jaszkiewicz, A., 2003. Multiple objective metaheuristics for combinatorial optimization: A tutorial. *In: Proceedings of the 4th Metaheuristic International Conference (MIC2003)*, 25-28.

TEI07 Teixeira, C., Faria, R., Covas, J. A. and Gaspar-Cunha, A., 2007. Modelling Flow and Heat Transfer in Co-Rotating Twin-Screw Extruders. *In:* E. Cueto and F. Chinesta, eds. *10th ESAFORM Conference on Material Forming. AIP Conference Proceedings*, Vol.907. Berlin: Springer Verlag, 980–985.

TEI10a Teixeira, C., Covas, J. A., Berzin, F., Vergnes B. and Gaspar-Cunha, A., 2010. Application of Evolutionary Algorithms to the Definition of the optimal Twin-Screw Extruder Configuration for Starch Cationization. *Polymer Engineering and Science*. 51 (2), 330–340.

TEI10b Teixeira, C., Covas, J. A., Stützle, T. and Gaspar-Cunha, A., 2010. Optimization of Co-rotating Twin-Screw Extruders Using Pareto Local Search. *In:* X. Gao, A. Gaspar-Cunha, M. Köppen, G. Schaefer and J. Wang, eds. *Soft Computing in Industrial Applications,* Vol. 75. Heidelberg: Springer, 3-10.

TEI11 Teixeira, C., Covas, J. A., Stützle, T. and Gaspar-Cunha, A., 2011. Engineering an Efficient Two-Phase Local Search for the Co-Rotating Twin-Screw Configuration Problem. *International Transactions in Operational Research*, 18 (2), 271-291.

TEI12a Teixeira, C., Covas, J. A., Stützle, T. and Gaspar-Cunha, A., 2012. Multi-objective ant colony optimization for the twin-screw configuration problem. *Engineering Optimization,* 44 (3), 351-371.

TEI12b Teixeira, C., Gaspar-Cunha, A. and Covas, J. A., 2012. Flow and heat transfer along the length of a co-rotating twin screw extruder. *Polymer Plastics Technology and Engineering*, 51 (15), 1567-1577.

TKI02 T'kindt, V., Monmarché, N., Tercinet, F. and Läugt, D., 2002. An Ant Colony Optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research,* 142 (2), 250–257.

TOD93 Todd, D. B., 1993. Melting of Plastics in Kneading Blocks. *International Polymer Processing*, 8 (2), 113-118.

ULD91 Ulder, N. L. J., Aarts, E. H. L. and Bandelt, H.-J., 1991. Genetic Local Search Algorithms for the Travelling Salesman Problem. *In:* H.-P. Schwefel and R. Männer, eds. *Parallel Problem Solving from Nature. Lecture Notes in Computer Science*, Vol. 496. Springer, 109-116.

ULU99 Ulungu, L. E., Teghem, J., Fortemps, P. H. and Tuyttens, D., 1999. MOSA method: A tool for Solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 8 (4), 221-236.

VAL09 Valette, R., Coupez, T., David, C. and Vergnes, B., 2009. A Direct 3D Numerical Simulation Code for Extrusion and Mixing Processes. *International Polymer Processing*, 24 (2), 141-147.

VAN96 Van Der Wal, D. J., Goffiw, D., Klomp, E. M., Hoogstraten, H. W. and Janssen, L. , 1996. Three-Dimensional Flow Modeling of a Self-wiping Corotating Twin-Screw Extruder - Part II: The Kneading Section. *Polymer Engineering and Science*, 36 (7), 912-924.

VER98 Vergnes, B., Della Valle, G. and Delamare, L., 1998. A Global Computer Software for Polymer Flows in Corotating Twin Screw Extruders. *Polymer Engineering and Science*, 38 (11), 1781-1792.

VER00 Vergnes, B. and Berzin, F., 2000. Peroxide-controlled degradation of polypropylene: rheological behaviour and process modeling. *Macromolecular. Symposia*, 158 (1), 77-90.

VER01 Vergnes, B., Souveton, G., Delacour, M. L. and Ainser, A., 2001. Experimental and Theoretical Study of Polymer Melting in a Co-rotating Twin Screw Extruder. *International Polymer Processing*, 16 (4), 351-362.

VER04 Vergnes, B. and Berzin, F., 2004. Modelling of flow and chemistry in twin screw extruders. *Plastics Rubber and Composites,* 33 (9), 409-415.

VOß01 Voß, S, 2001. Meta-heuristics: The state of the art. *In:* A. Nareyek, ed. *Local Search for Planning and Scheduling. Lectures Notes in Artificial Intelligence*, Vol.2148. Berlin: Springer, 1-23.

VOß09 Voß, S., 2009. Metaheuristics. *In:* C. A. Floudas and P. M. Pardalos, eds. *Encyclopedia of Optimization*. 2nd edition. New York: Springer, 2061-2075.

WAN89a Wang, Y. and White, J. L., 1989. Non-Newtonian Flow Modeling in the Screw Regions of an Intermeshing Corotating Twin Screw Extruder. *Journal of Non-Newtonian Fluid Mechanics*, 32 (1), 19-38.

WAN89b Wang, Y., White, J. L. and Szydlowski, W., 1989. Flow in a Modular Intermeshing Corotating Twin Screw Extruder. *International Polymer Processing*, 4 (4), 262-269.

WER79 Werner, H. and Eise, K., 1979. An Analysis of the Conveying Characteristics of Twin Screw Co-rotating extruders. *SPE Antec Technical Papers*, 37, 181-187.

WHI87 White, J. L. and Szydlowski, W., 1987. Composite Models of Modular Intermeshing Corotating and Tangential Counter-Rotating Twin Screw Extruders. *Advances in Polymer Technology*, 7 (4), 419-426.

WHI01a White, J. L., Kim, E. K., Keum, J. M., Jung, H. C. and Bang, D. S., 2001. Modeling Heat Transfer in Screw Extrusion with Special Application to Modular Self-Wiping Co-Rotating Twin Screw Extrusion. *Polymer Engineering and Science*, 41 (8), 1448-1455.

WHI01b White, J. L., Kim, B., Bawiskar, S. and Keum, J. M., 2001. Development of a Global Computer Software for Modular Self-Wiping Corotating Twin Screw Extruders. *Polymer-Plastics Technology and Engineering*, 40 (4), 385-405.

WHI01c White, J., Montes, S., Szydlowski, W., Chen, Z. and Bawiskar, S., 2001. Co-Rotating Twin Screw Extruder Modeling Program [online]. Available from: http://www.temarex.com/extrus.htm [Accessed 28 January 2011].

WHI10 White, J. L. and Kim, E. K., 2010. *Twin Screw Extrusion; Technology and Principles.* 2nd ed. Munich: Hanser.

WIL12 Wilczynski, K., Nastaj, A., Lewandowski, A. and Wilczynski, K.J., 2012. Multipurpose Computer Model for Screw Processing of Plastics. *Polymer-Plastics Technology and Engineering,* 51 (6), 626-633.

WON97 Wong, A. C.-Y., Zhu, F. and Liu, T., 1997. Qualitative study on intermeshing co-rotating twin screw extrusion using novel visual technique. *Plastics, Rubber and Composites Processing and Applications*, 26 (6), 271-277.

WON00 Wong, A. C.-Y., Zhu, F. and Liu, T., 2000. Solid Transportation in the Feeding Zone of Intermeshing Co-rotating Twin-screw Extruders. *Journal of Polymer Research*, 7 (3), 135-147.

XU06 Xu, J., Sohoni, M., McCleery, M. and Bailey, T. G., 2006. A dynamic neighborhood based tabu search algorithm for real-world flight instructor scheduling problems. *European Journal of Operational Research*, 169 (3), 978-993.

YAN92 Yang, H. H. and Manas-Zloczower, I., 1992. Flow Field Analysis of the Kneading Disc Region in a Co-rotating Twin Screw Extruders, *Polymer Engineering and Science*, 32 (19), 1411-1417.

YIC03 Yichong, G. and Fuhua, Z., 2003. Study of Different Flow Patterns in Melting Section of a Co-Rotating Twin-Screw Extruder. *Polymer Engineering and Science*, 43 (2), 306-316.

YOS00 Yoshinaga, M., Katsuki, S., Miyazaki, M., Liu, L., Kihara, S. and Funatsu, K., 2000. Mixing mechanism of Three-Tip Kneading Block in Twin Screw Extruders, *Polymer Engineering and Science*, 40 (1), 168-178.

YOS01 Yoshinaga, M., Kihara, S. and Funatsu, K., 2001. 3-D Non-Isothermal Flow Field Analysis and Mixing Performance Evaluation of Kneading Blocks in a Co-rotating Twin Screw Extruder. *Polymer Engineering and Science*, 41 (5), 840-849.

ZAG05 Zagal, A., Vivaldo-Lima, E. and Manero, O., 2005. A Mathematical Model for the Reactive Extrusion of Methyl Methacrylate in a Co-Rotating Twin Screw Extruder. *Industrial & Engineering Chemical Research,* 44 (26), 9805-9817.

ZHU01 Zhu, L., Narh, K. A. and Geng, X., 2001. Modeling of Particle-Dispersed Melting Mechanism and its Application on Corotating Twin-Screw Extrusion. *Journal of Polymer Science: Part B*, 39 (20), 2462-2468.

ZHU05 Zhu, L., Narh, K. A. and Hyun, K. S., 2005. Evaluation of numerical simulation methods in reactive extrusion. *Advances in Polymer Technology,* 24 (3), 183-193.

ZIL96 Zilberstein, S., 1996. Using Anytime Algorithms in Intelligent Systems. *AI Magazine*, 17 (3), 73–83.

ZIT99 Zitzler, E. and Thiele, L., 1999. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation,* 3 (4), 257-271.

ZIT01 Zitzler, E., Laumanns, M. and Thiele, L., 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *In*: K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty, eds. *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN2001).* International Center for Numerical Methods in Engineering, 95-100.

ZIT03 Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M. and Grunert da Fonseca, V., 2003. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7 (2), 117–132.

ZIT08 Zitzer, E., Knowles, J. and Thiele, L., 2008. Quality Assessment of Pareto Set Approximations. *In:* J. Branke, K. Deb, K. Miettinen and R. Slowinski, eds. *Multiobjective Optimization: Interactive and Evolutionary Approaches. Lecture Notes in Computer Science,* Vol. 5252, 373-404.

Appendix A

Table A.1. Comparison of the solution quality obtained by the 2-swap operator and by the others operators with first and best improvement rules for TSCP4. The average differences were obtained through the t-test. Concerning the difference, if the value is negative (positive), the 2-swap operator obtained a solution quality that is smaller (larger) than the one obtained by the other operator considered. In both cases, the value is typeset in italics if it is significantly different from zero according to the t-test, at a confidence level of 95%.

First Improvement Rule			Best Improvement Rule		
Comparison	**Avg diff (%)**	**95% CI**	**Comparison**	**Avg diff (%)**	**95% CI**
Maximization of average strain			*Maximization of average strain*		
2S vs IF	*+2.447*	[+0.783, +4.111]	2S vs IF	*+3.660*	[+1.841, +5.480]
2S vs IB	*+1.987*	[+0.726, +3.247]	2S vs IB	*+2.705*	[+1.117, +4.294]
2S vs 2E	*+4.452*	[+2.740, +6.165]	2S vs 2E	*+5.649*	[+3.508, +7.791]
2S vs AS	*+10.090*	[+8.080, +12.101]	2S vs AS	*+11.191*	[+9.110, +13.272]
Minimization of SME			*Minimization of SME*		
2S vs IF	*-1.658*	[-2.508, -0.808]	2S vs IF	*-2.425*	[-3.318, -1.533]
2S vs IB	*-1.649*	[-2.700, -0.598]	2S vs IB	*-2.140*	[-3.132, -1.148]
2S vs 2E	*-2.265*	[-3.384, -1.146]	2S vs 2E	*-1.811*	[-2.719, -0.903]
2S vs AS	*-8.357*	[-9.680, -7.034]	2S vs AS	*-9.661*	[-10.995, -8.327]
Minimization of viscous dissipation			*Minimization of viscous dissipation*		
2S vs IF	*-0.691*	[-1.017, -0.364]	2S vs IF	*-1.739*	[-2.526, -0.951]
2S vs IB	*-0.778*	[-1.245, -0.310]	2S vs IB	*-0.795*	[-1.257, -0.332]
2S vs 2E	*-1.037*	[-1.358, -0.716]	2S vs 2E	*-0.424*	[-0.650, -0.199]
2S vs AS	*-3.913*	[-4.454, -3.371]	2S vs AS	*-3.945*	[-4.538, -3.351]

Table A.2. Comparison of the solution quality obtained by the sequential and restrictive best search strategies for the 12 case studies considered in this study. The average differences were obtained through the t-test. Concerning the difference, if the value is negative (positive), the sequential strategy obtained a solution quality that is smaller (larger) than the one obtained by the restrictive best strategy. In both cases, the value is typeset in italics if it is significantly different from zero according to the t-test, at a confidence level of 95%.

Instance	**Avg. diff (%)**	**95% CI**
Maximization of average strain		
TSCP1	*-2.509*	[- 4.763, - 0.256]
TSCP2	-1.837	[-3.911, + 0.237]
TSCP3	-1.247	[-3.567, + 1.072]
TSCP4	+0.536	[-3.753, + 4.824]
Minimization of SME		
TSCP1	+0.119	[-0.412, + 0.649]
TSCP2	+0.221	[-7.147, + 7.588]
TSCP3	+5.434	[+0.496, +10.372]
TSCP4	-1.271	[-2.670, + 0.129]
Minimization of viscous dissipation		
TSCP1	+0.049	[+0.004, + 0.093]
TSCP2	+0.176	[-0.042, + 0.394]
TSCP3	*+0.283*	[+0.068, + 0.498]
TSCP4	-0.000	[-0.133, + 0.133]

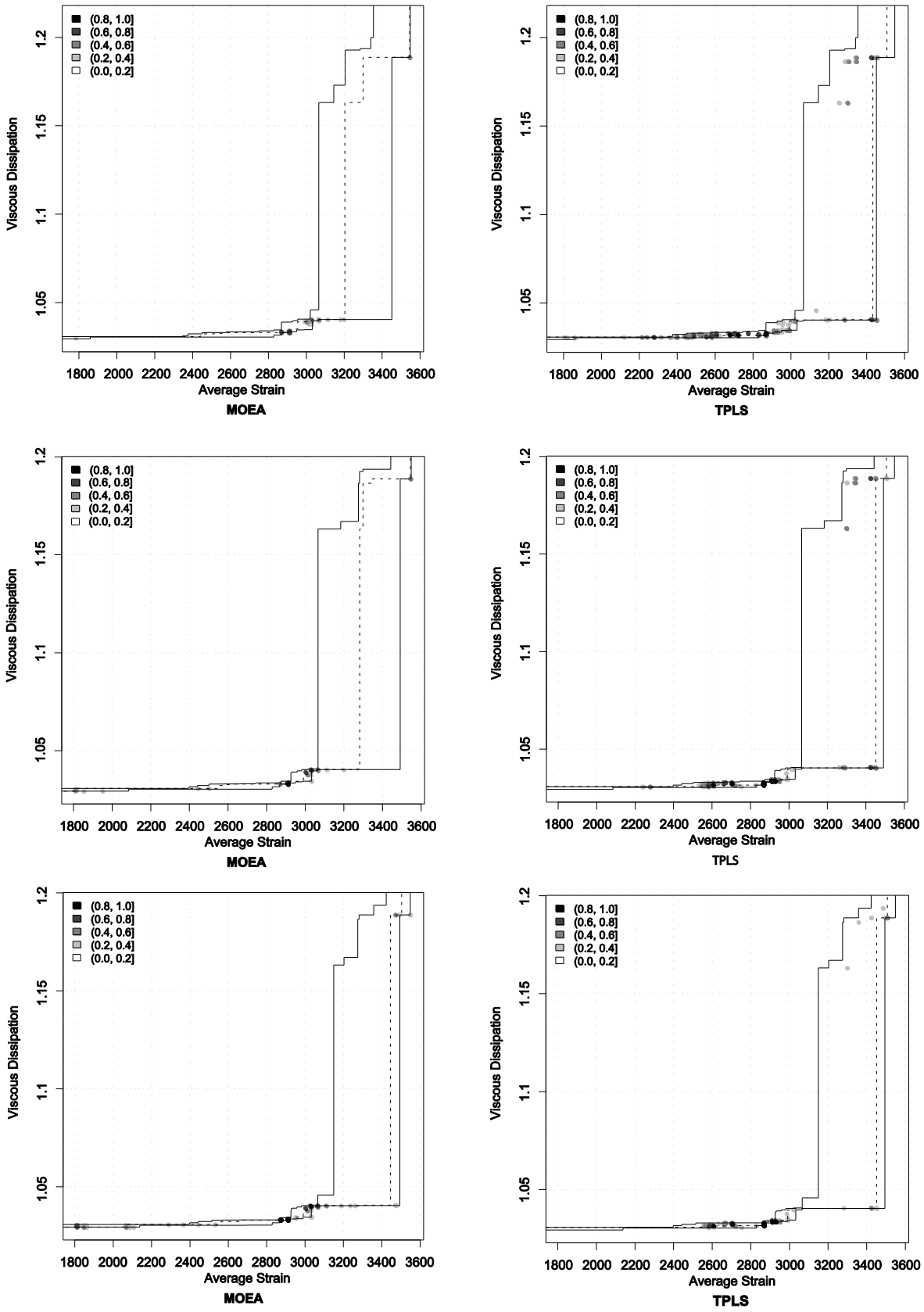

Figure A.1. Results for instance TSCP1 and case study two. Differences in terms of empirical attainment functions between the multi-objective evolutionary algorithm (MOEA) and two-phase local search (TPLS) after 1000 (top), 2000 (middle), and 3000 (bottom) evaluations of the simulation program for case study two. Advantages in favor of MOEA are indicated on the left side; those in favor of TPLS on the right side.

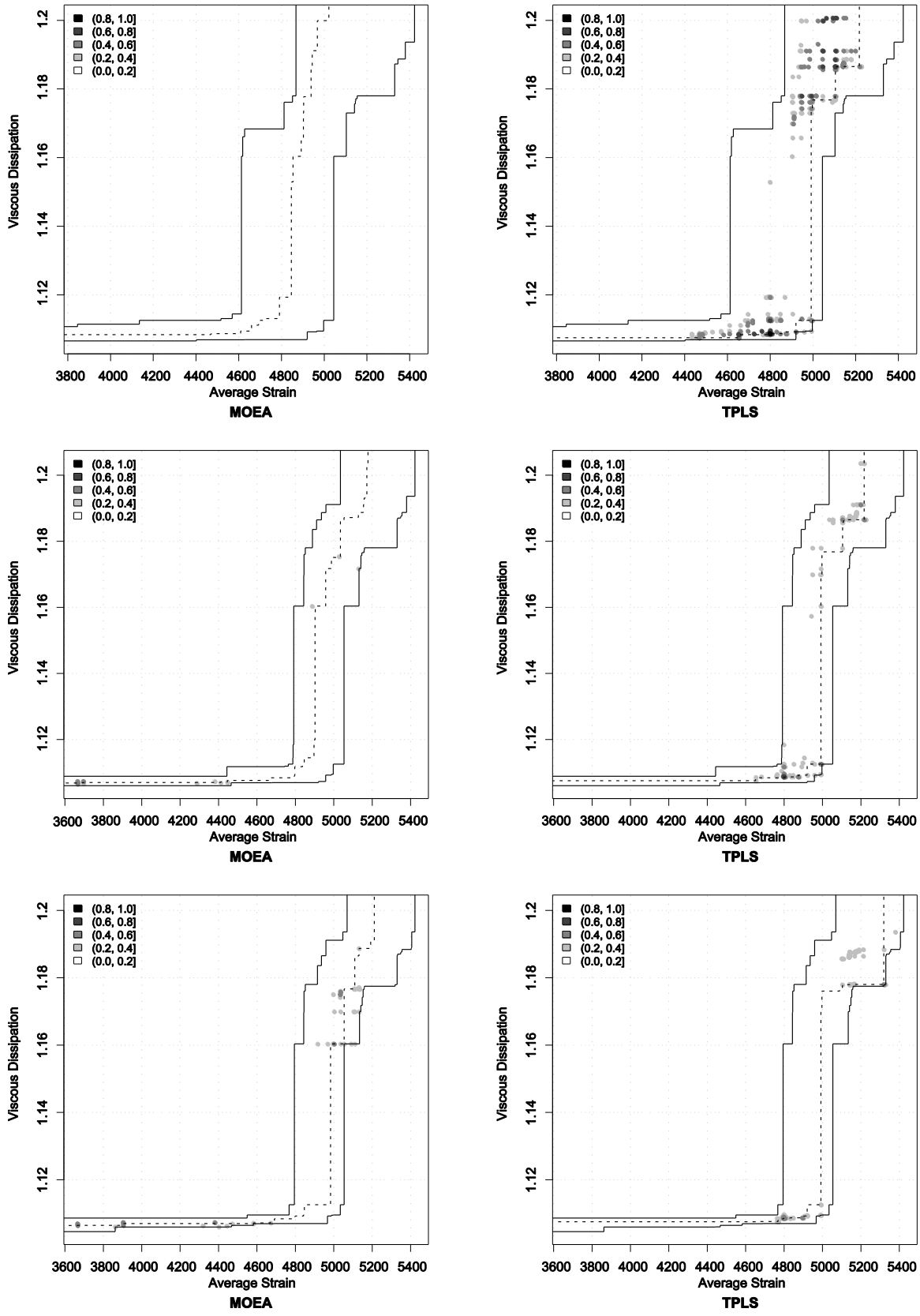

Figure A.2. Results for instance TSCP2 and case study two (see the caption of Figure A.1 for more details).

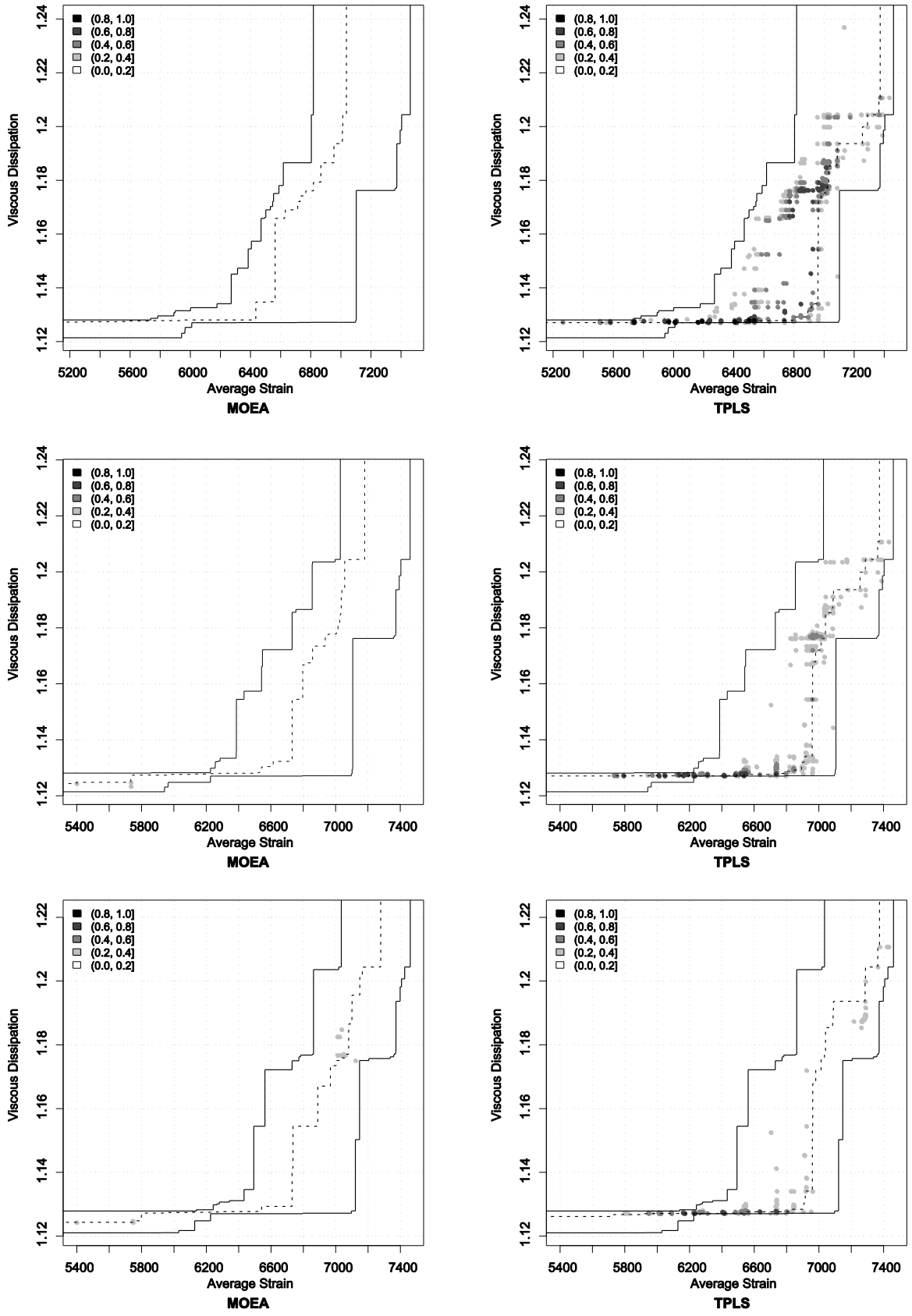

Figure A.3. Results for instance TSCP3 and case study two (see the caption of Figure A.1 for more details).

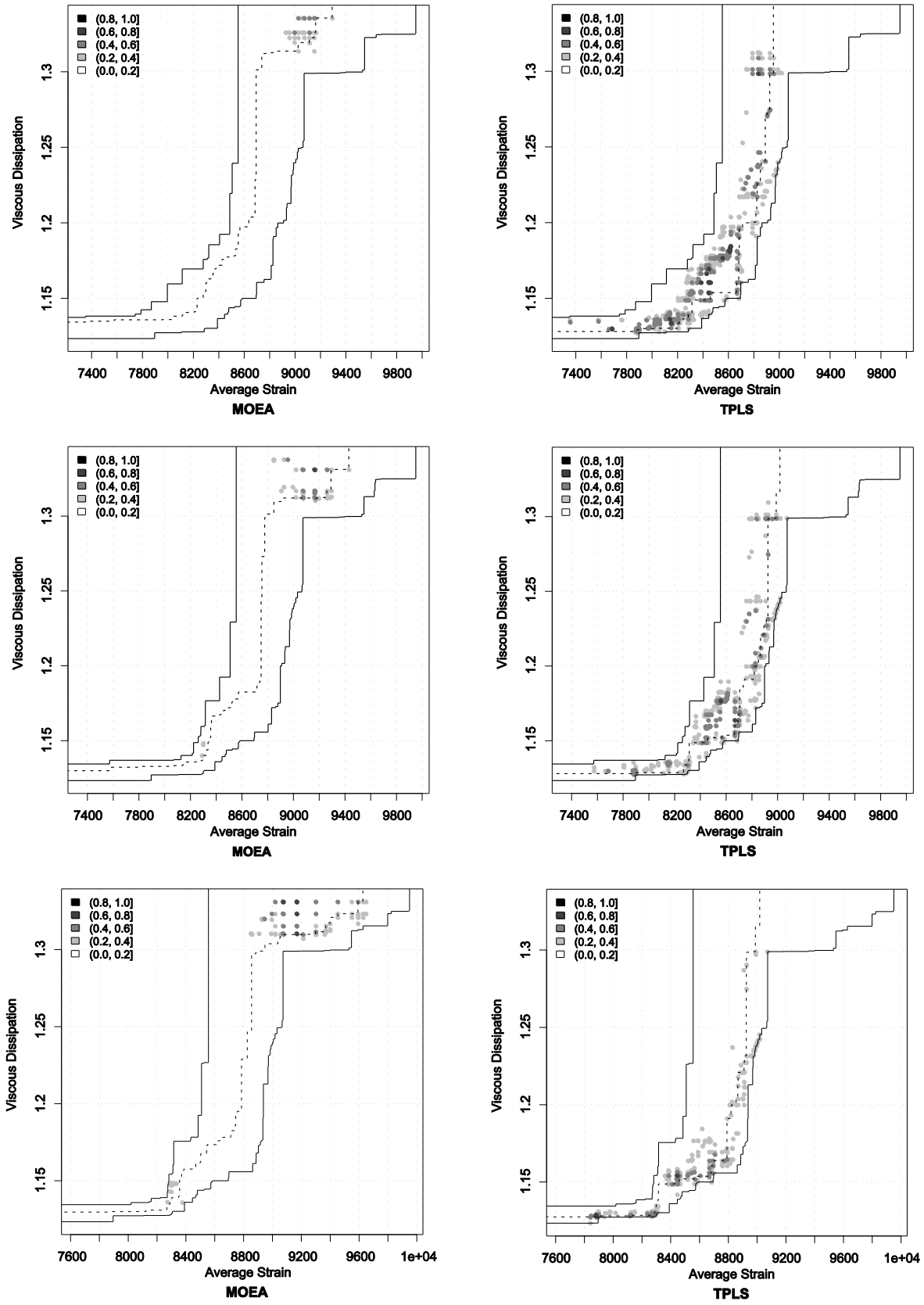

Figure A.4. Results for instance TSCP4 and case study two (see the caption of Figure A.1 for more details).

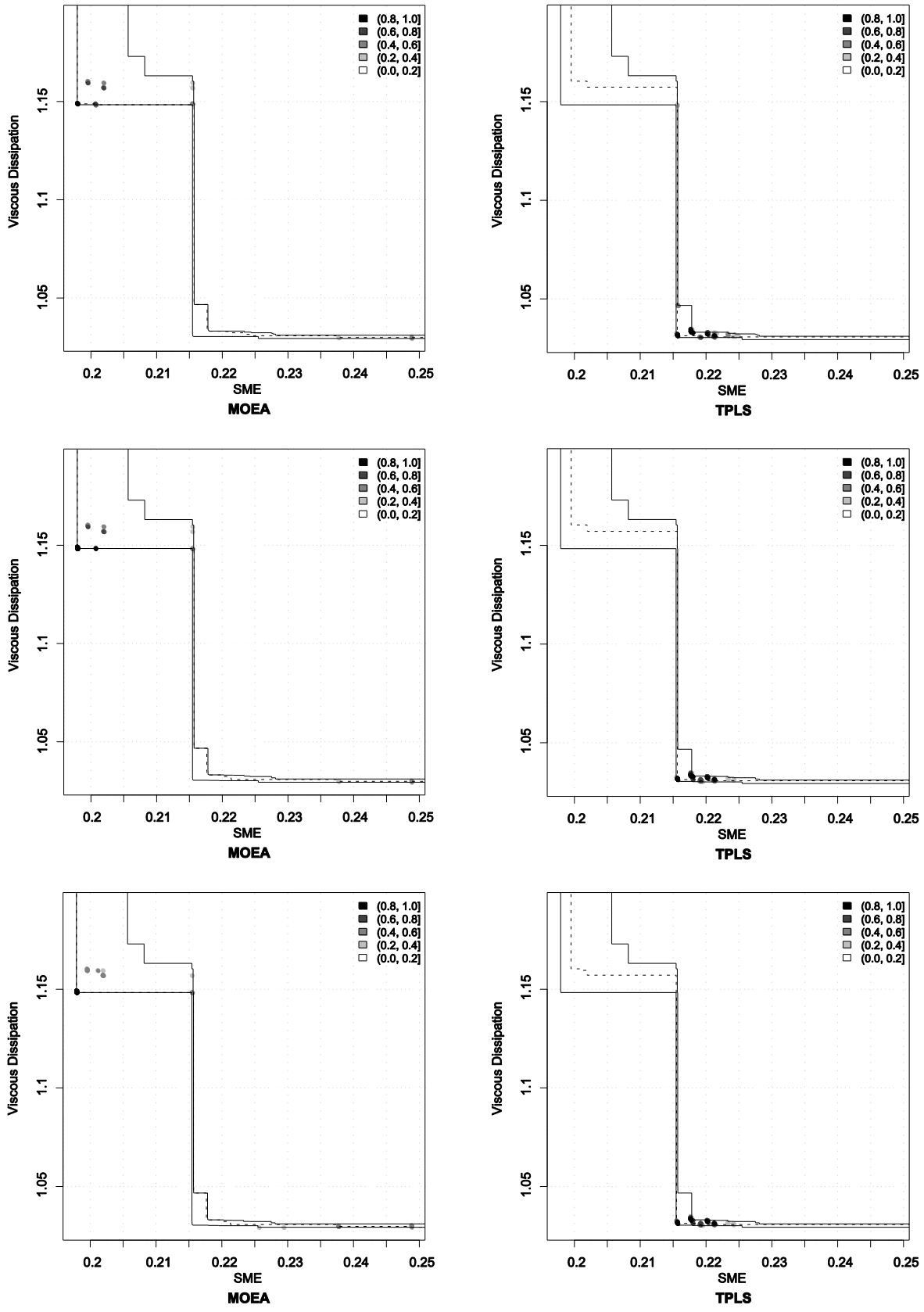

Figure A.5. Results for instance TSCP1 and case study three (see the caption of Figure A.1 for more details).

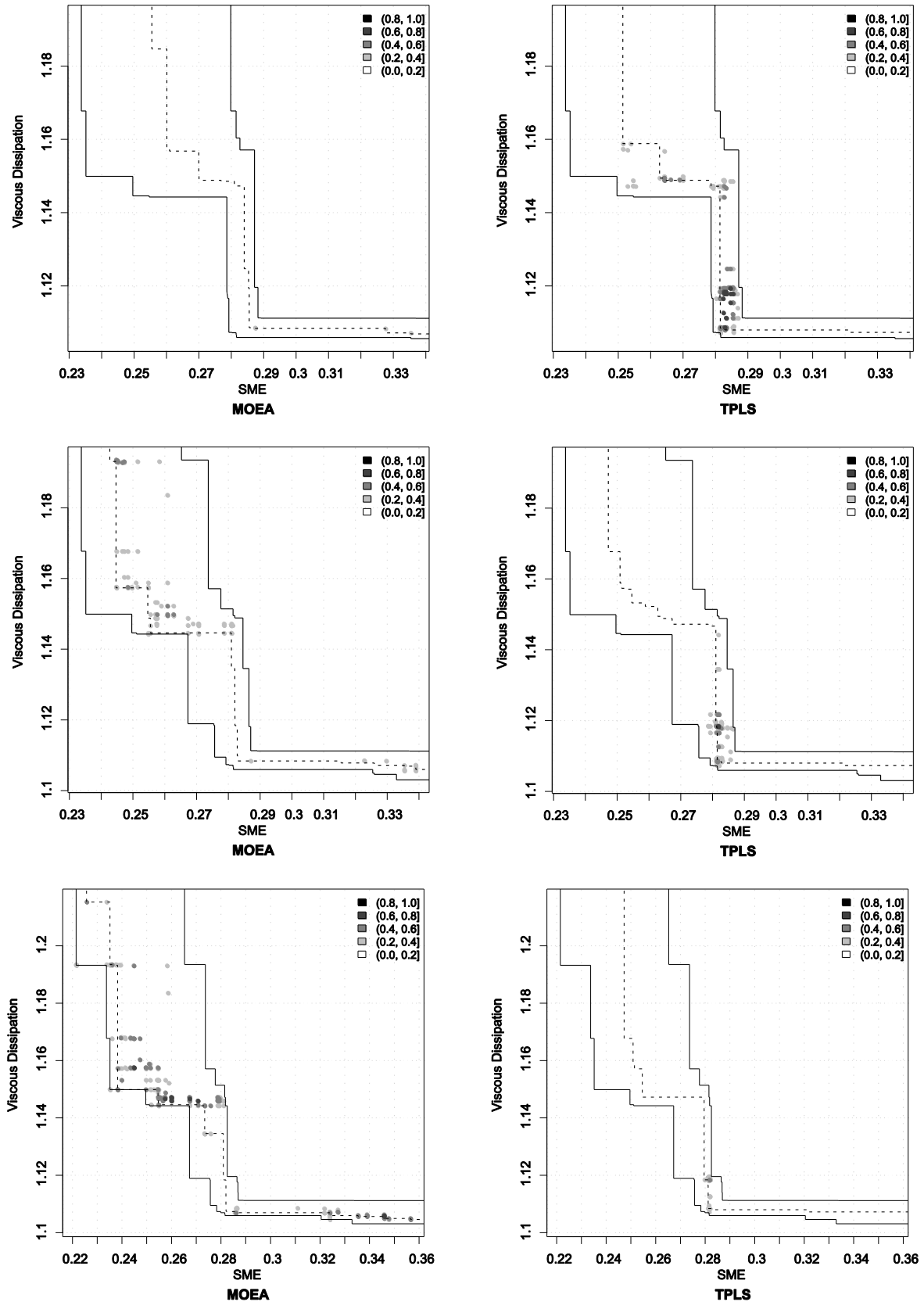

Figure A.6. Results for instance TSCP2 and case study three (see the caption of Figure A.1 for more details).

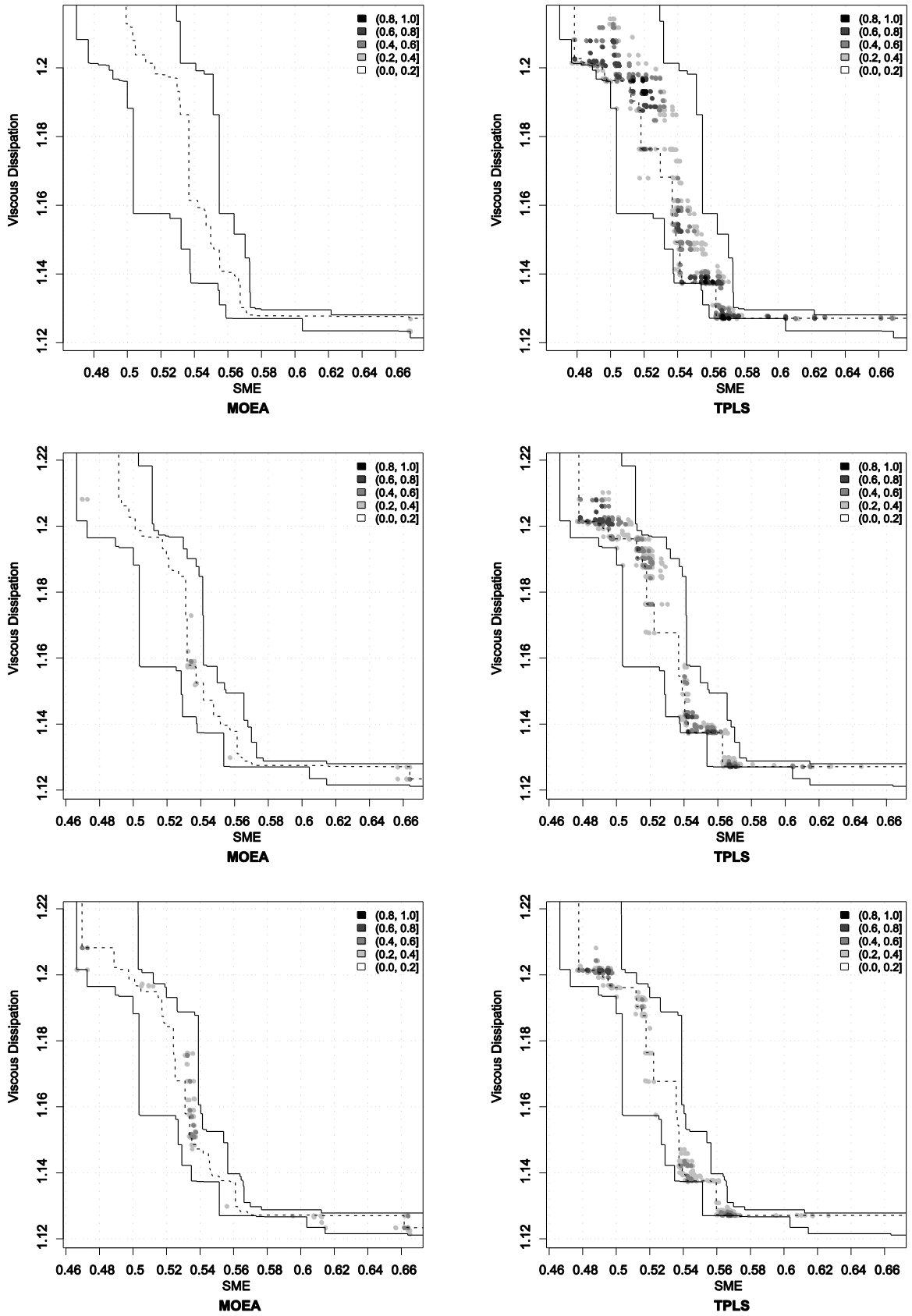

Figure A.7. Results for instance TSCP3 and case study three (see the caption of Figure A.1 for more details).

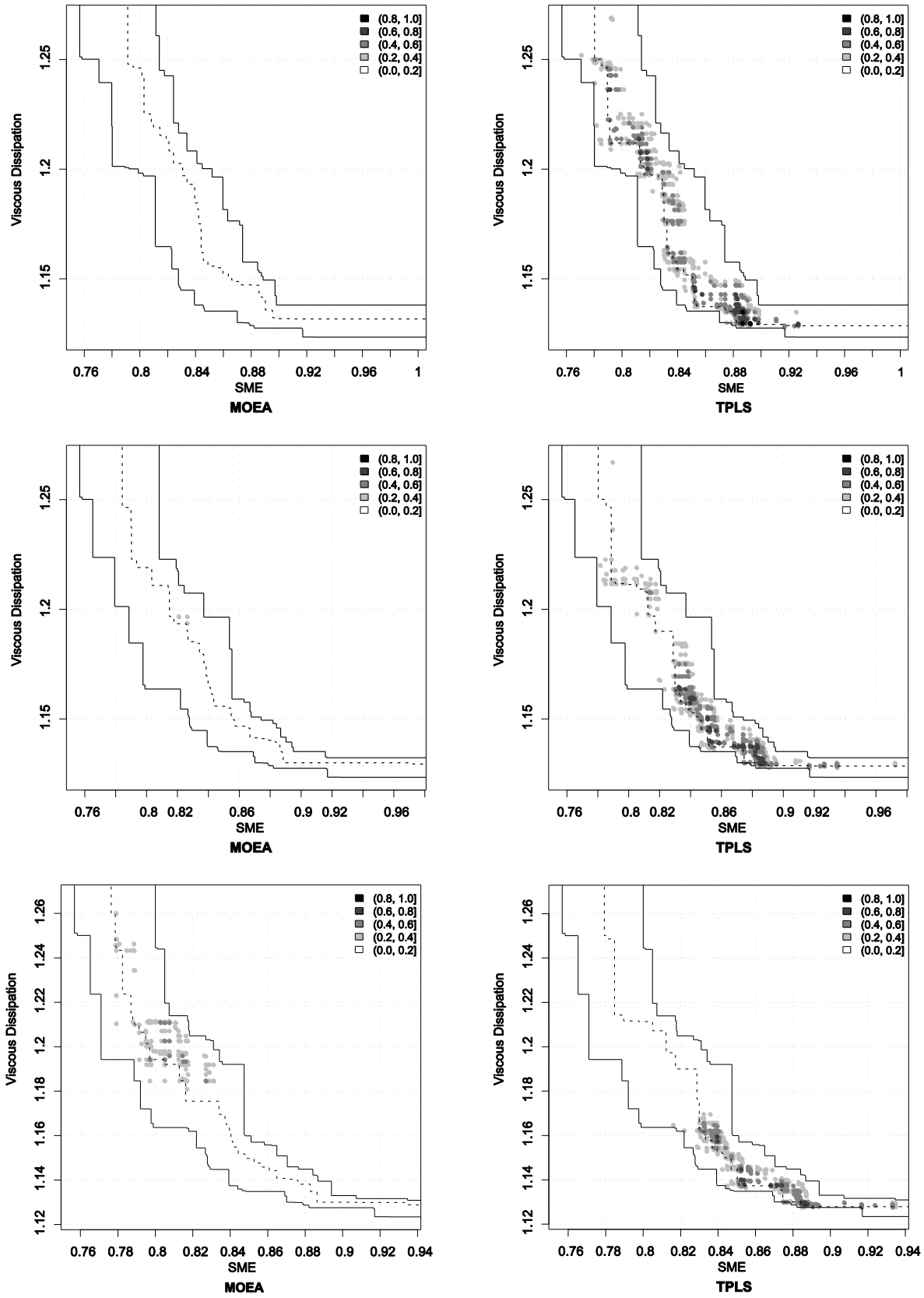

Figure A.8. Results for instance TSCP4 and case study three (see the caption of Figure A.1 for more details).

Appendix B

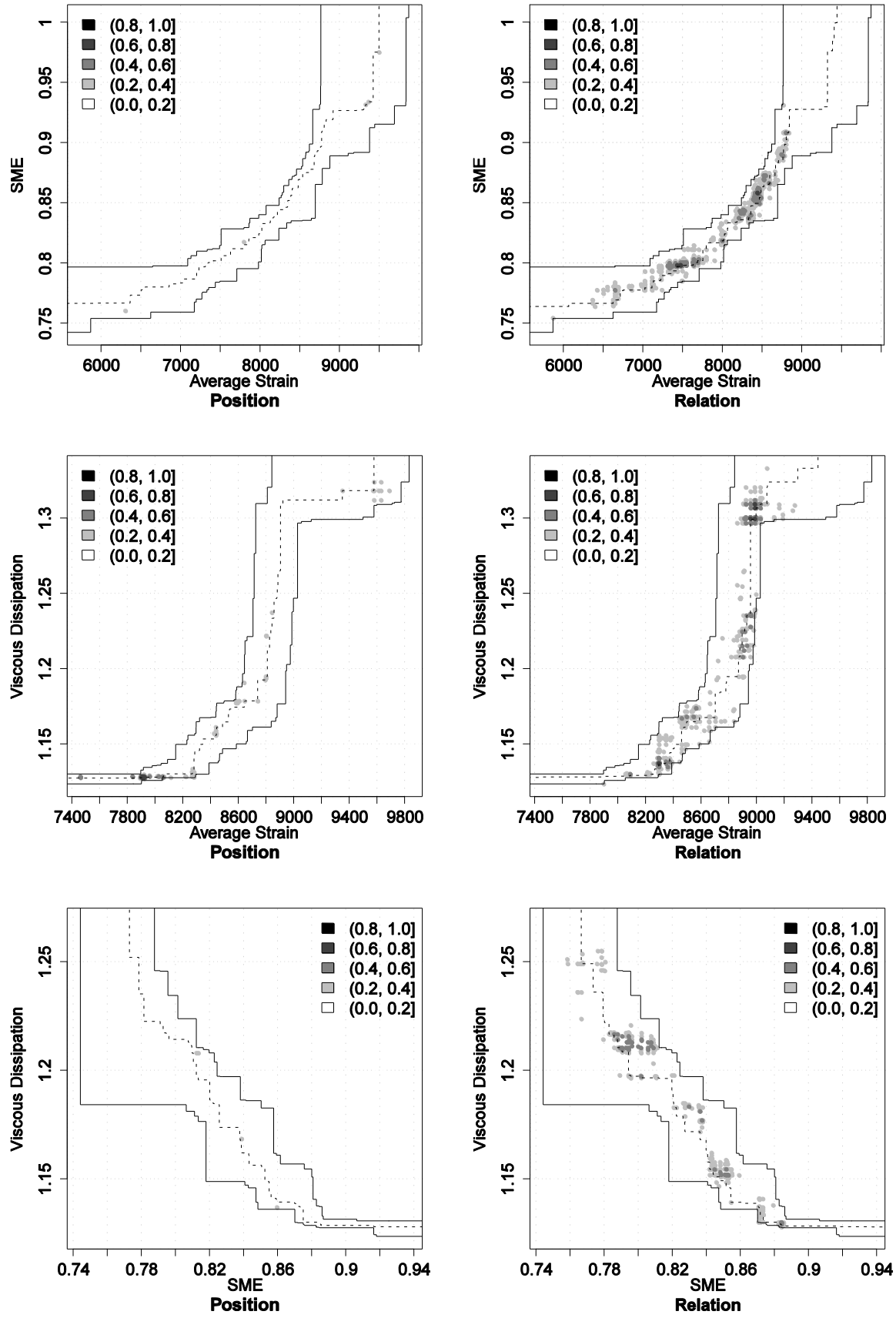

Figure B.1. Influence of pheromone information for TSCP4 in terms of empirical attainment functions. Advantages in favor of the position-based pheromone information are indicated on the left side; those in favor of relation-based pheromone information on the right side.

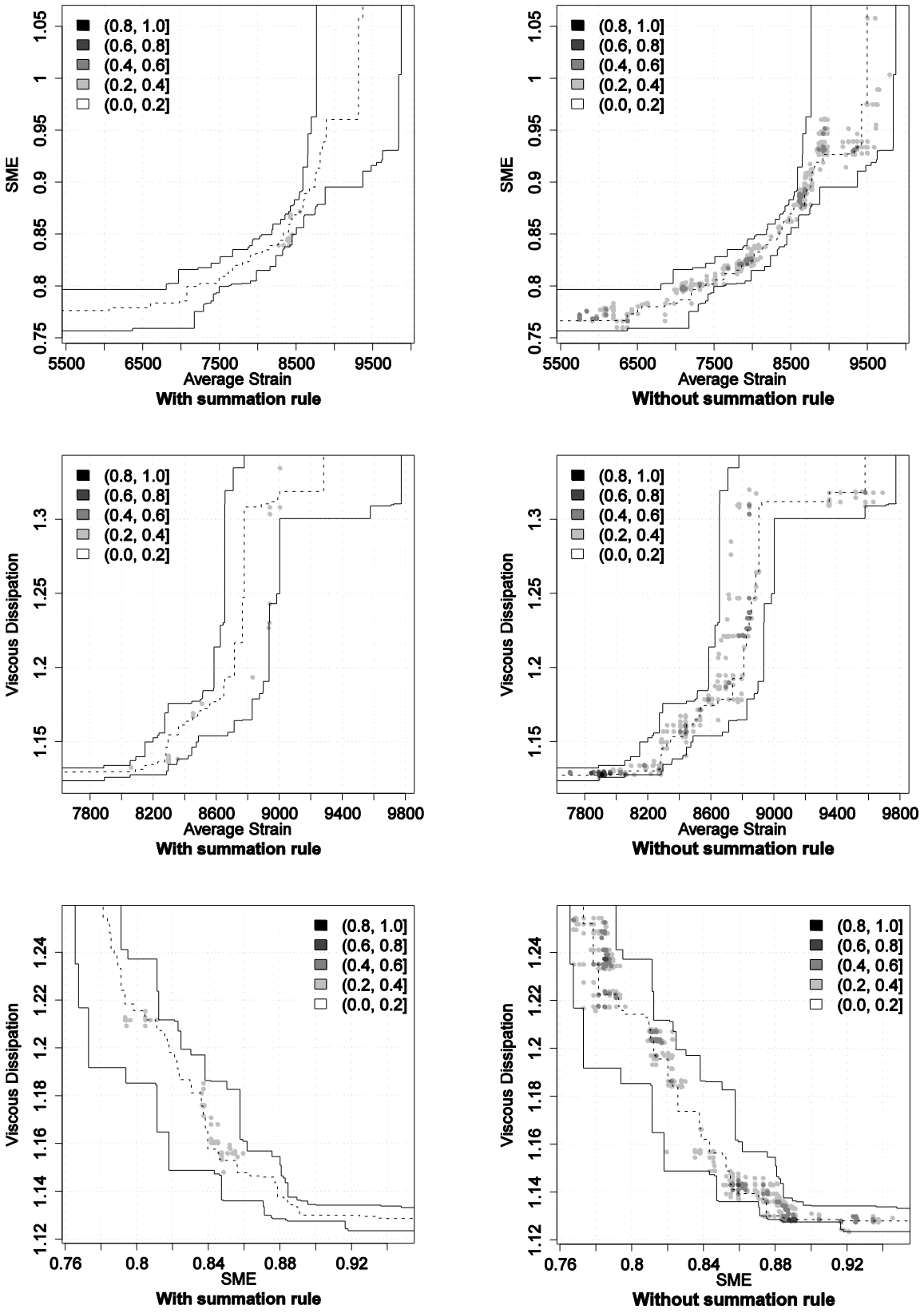

Figure B.2. Influence of the application of the pheromone summation rule for TSCP4 in terms of empirical attainment functions. Advantages in favor of the application of the pheromone summation rule are indicated on the left side; those in favor of no application of pheromone summation rule on the right side.

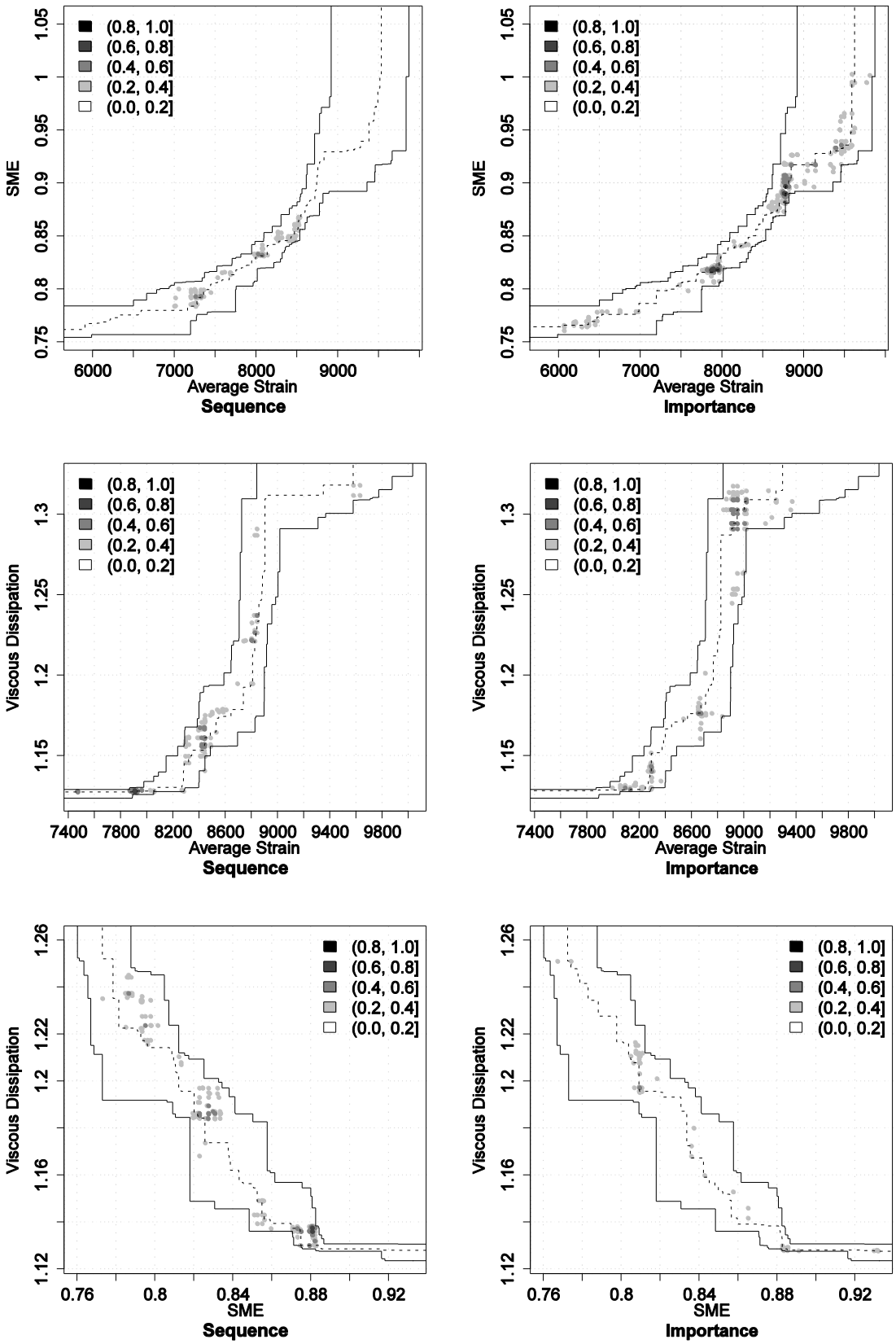

Figure B.3. Influence of assignment order of the screw elements for TSCP4 in terms of empirical attainment functions. Advantages in favor of sequential-based assignment order (with position-based pheromone information) are indicated on the left side; those in favor of importance-based assignment order on the right side.

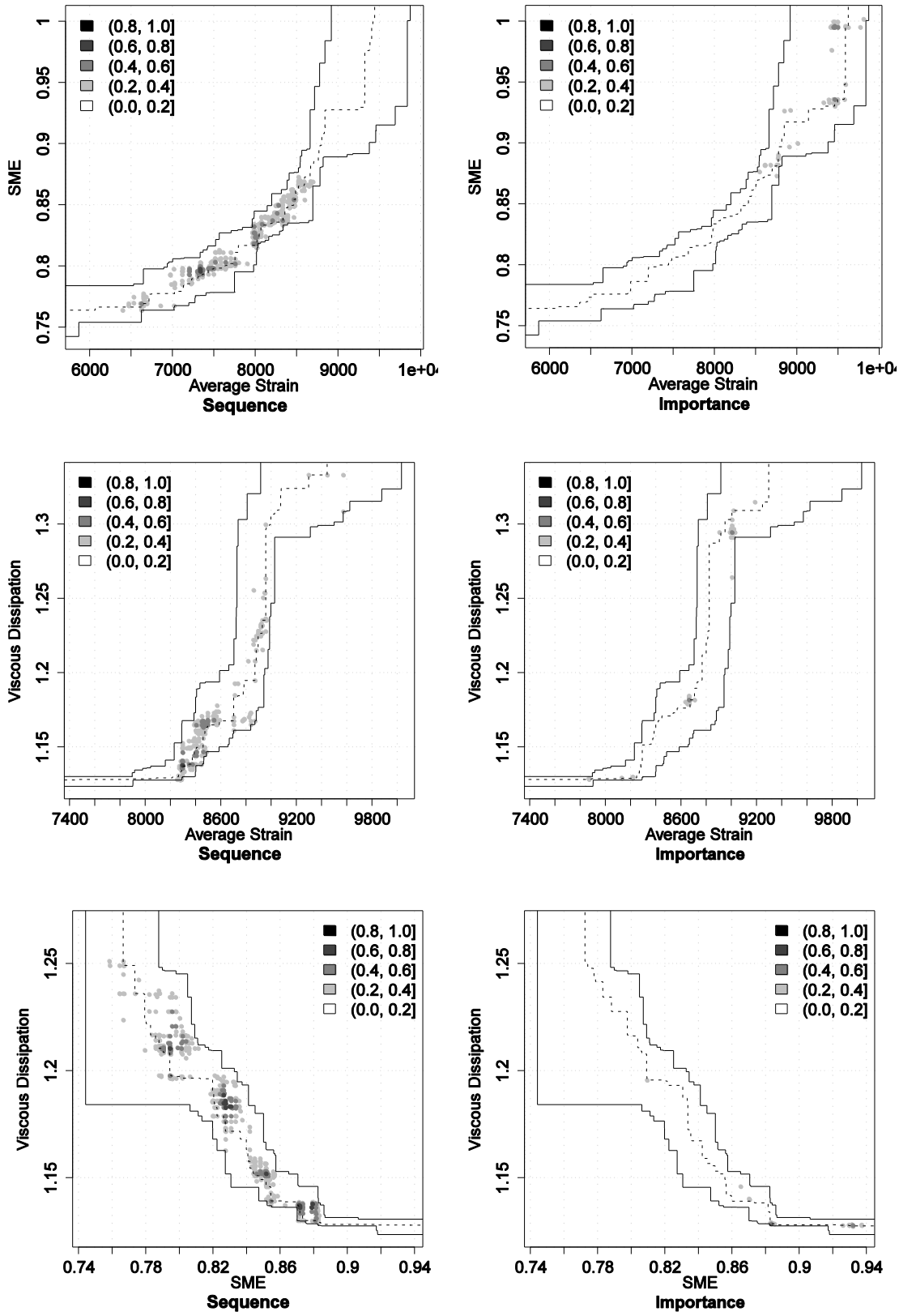

Figure B.4. Influence of assignment order of the screw elements for TSCP4 in terms of empirical attainment functions. Advantages in favor of sequential-based assignment order (with relation-based pheromone information) are indicated on the left side; those in favor of importance-based assignment order on the right side.

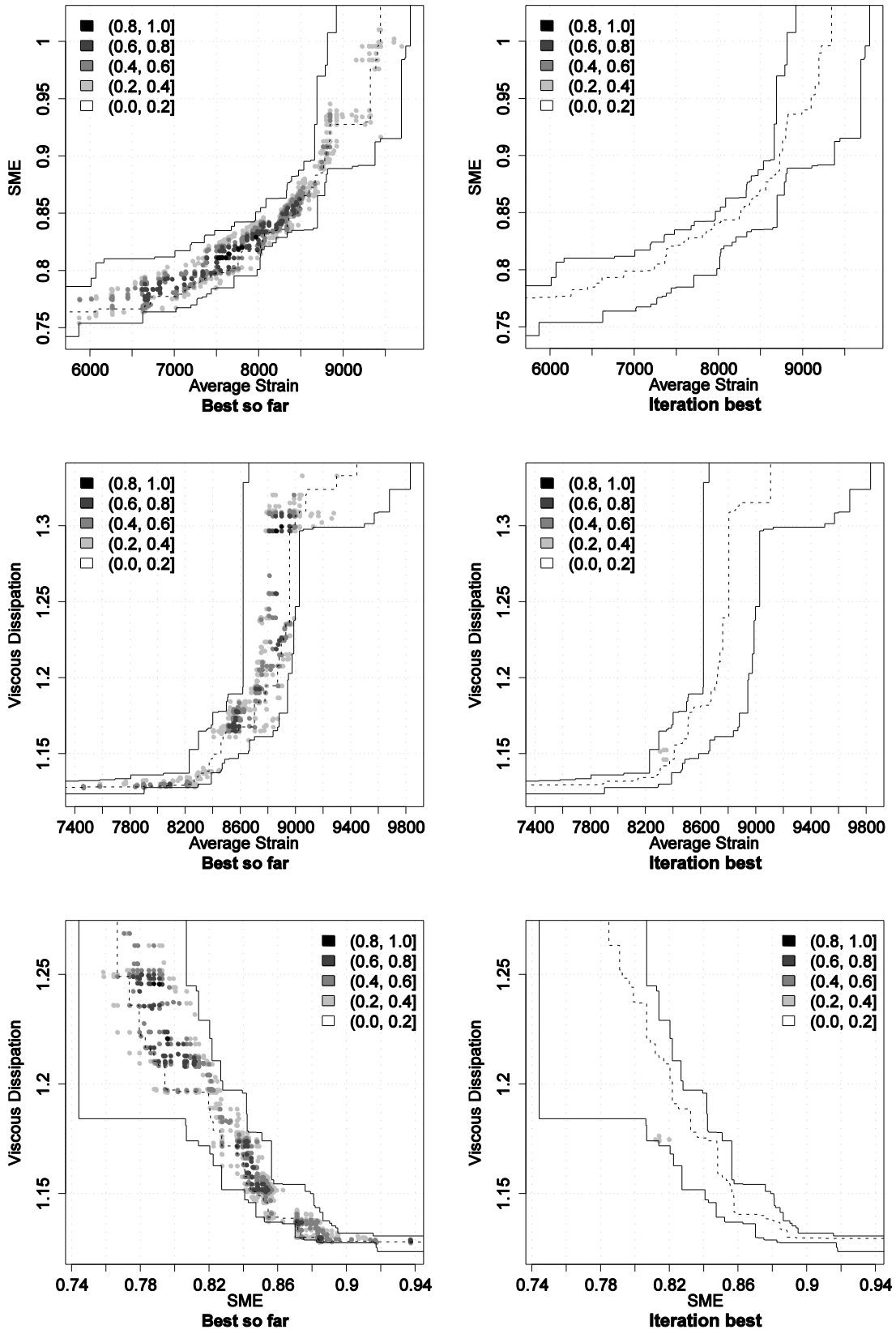

Figure B.5. Influence of the pheromone update strategy for TSCP4 in terms of empirical attainment functions. Advantages in favor of best-so-far strategy are indicated on the left side; those in favor of iteration best strategy on the right side.

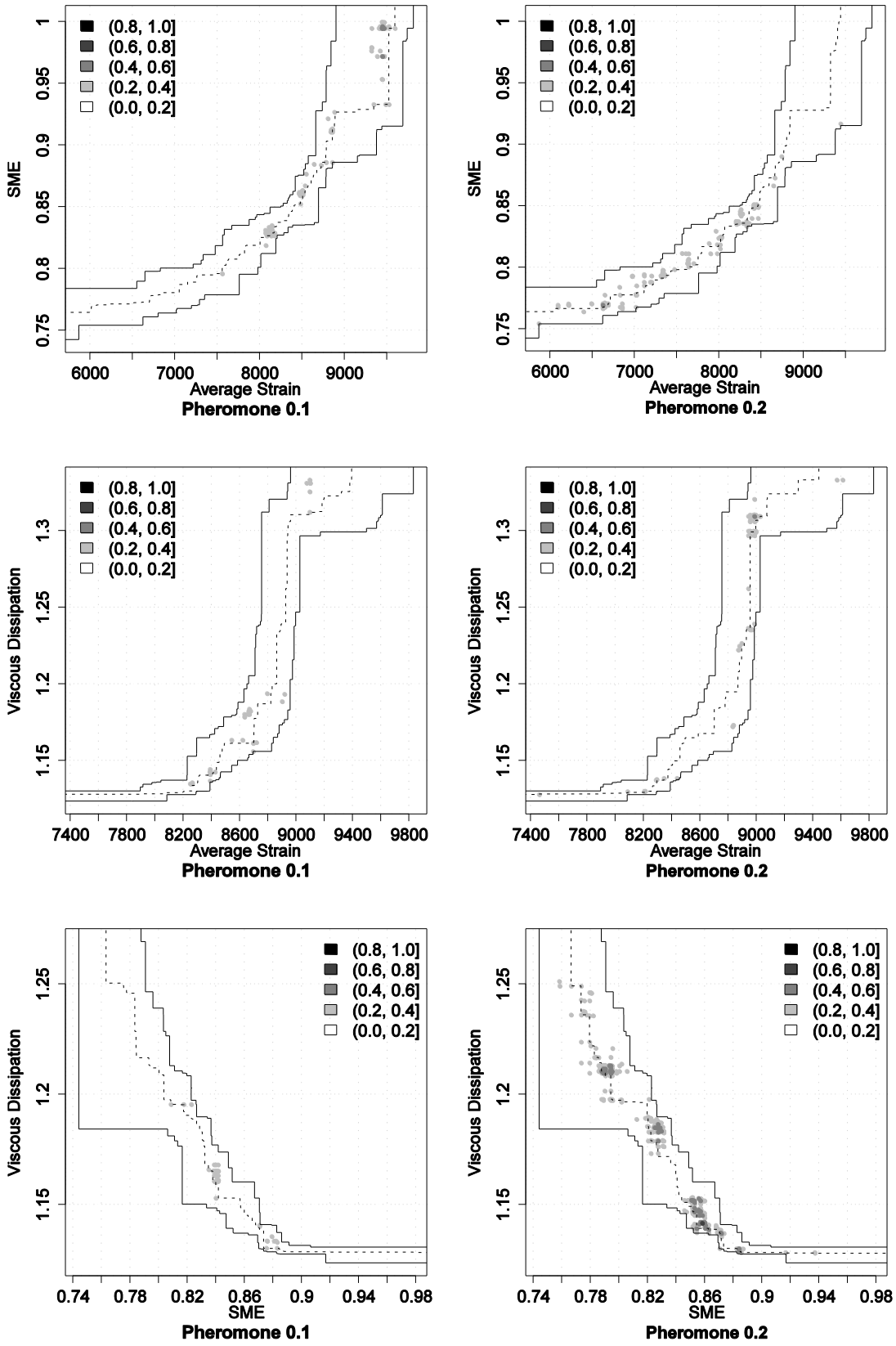

Figure B.6. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.1$ are indicated on the left side; those in favor of $\rho = 0.2$ on the right side.

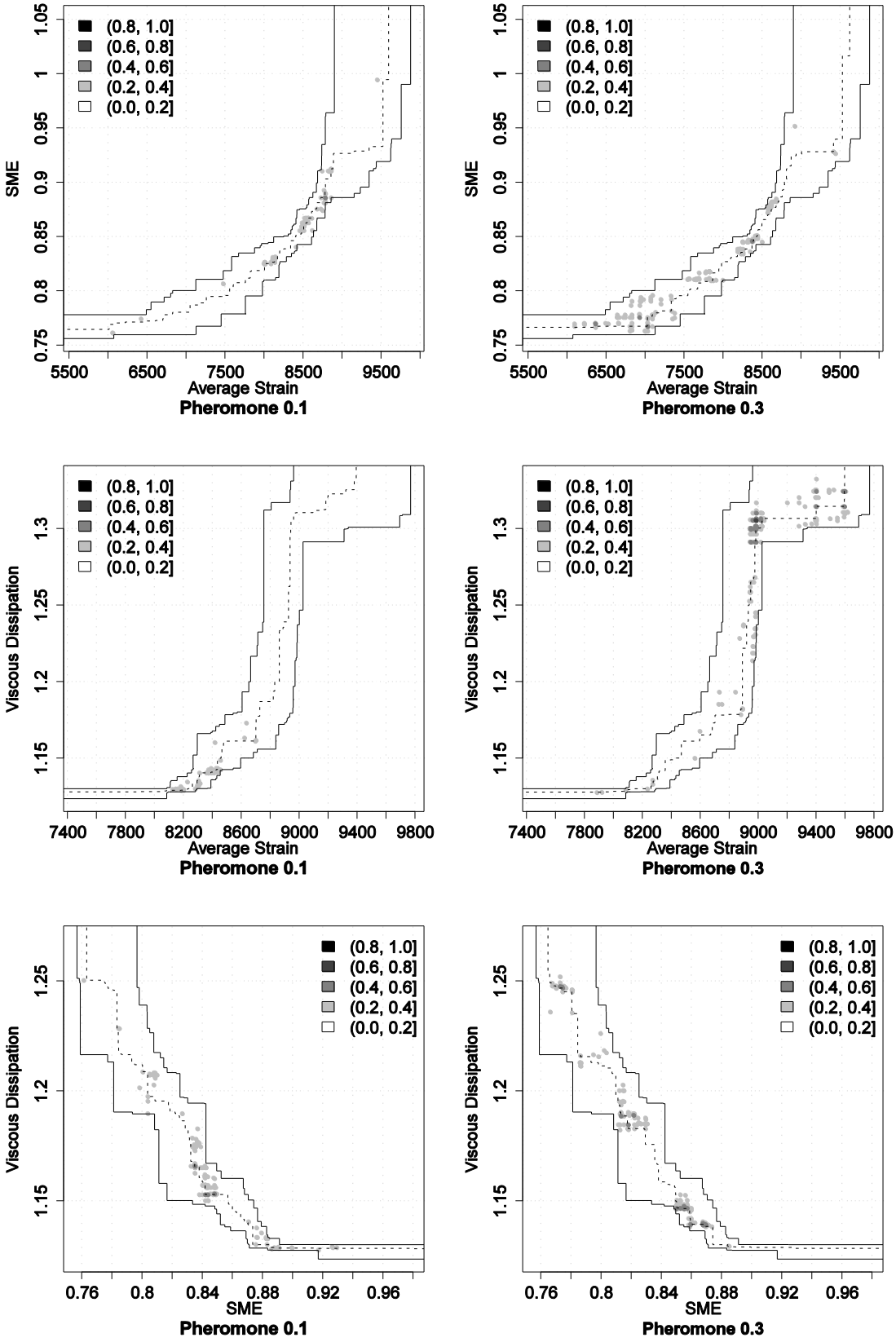

Figure B.7. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.1$ are indicated on the left side; those in favor of $\rho = 0.3$ on the right side.

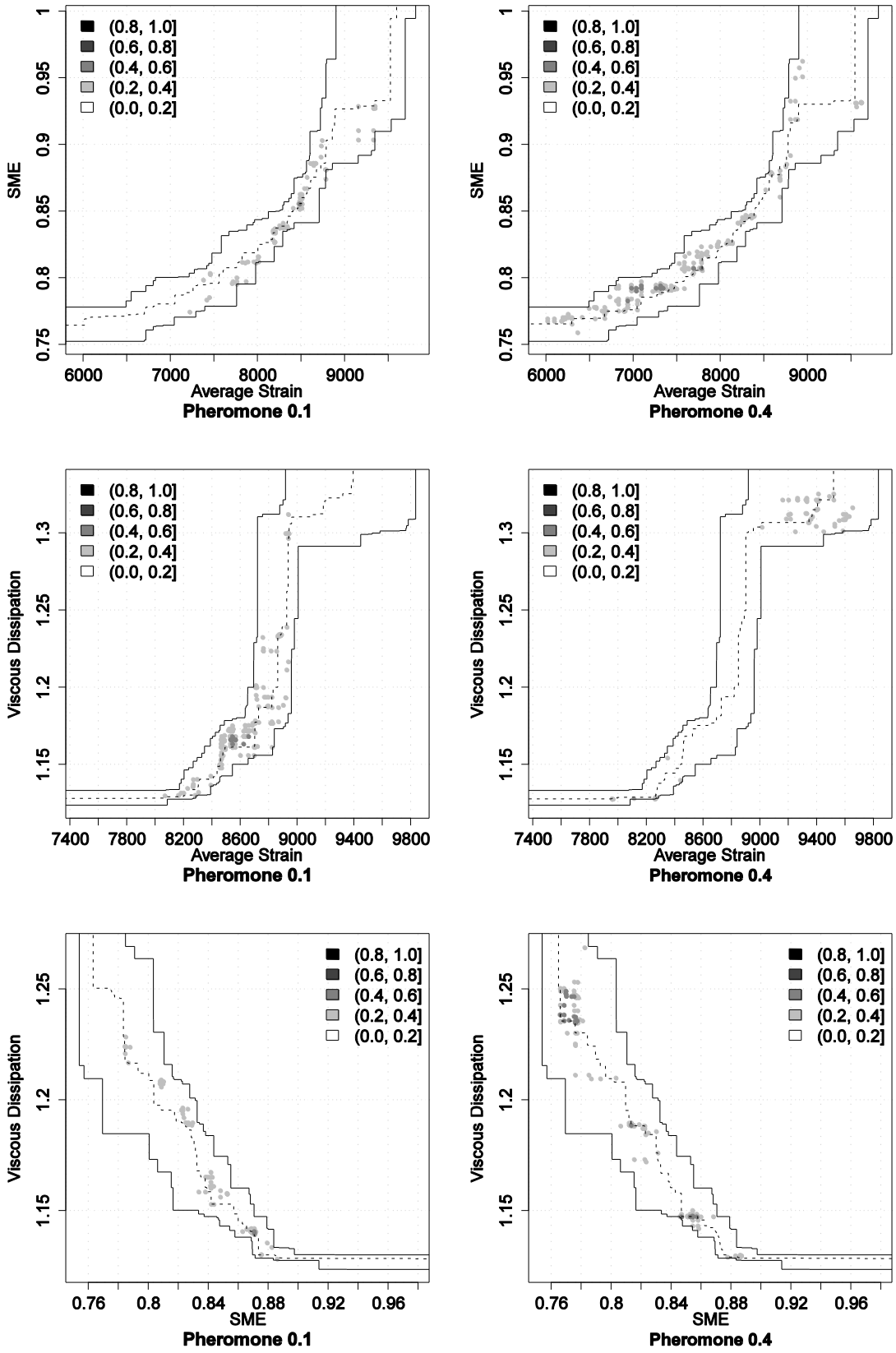

Figure B.8. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.1$ are indicated on the left side; those in favor of $\rho = 0.4$ on the right side.

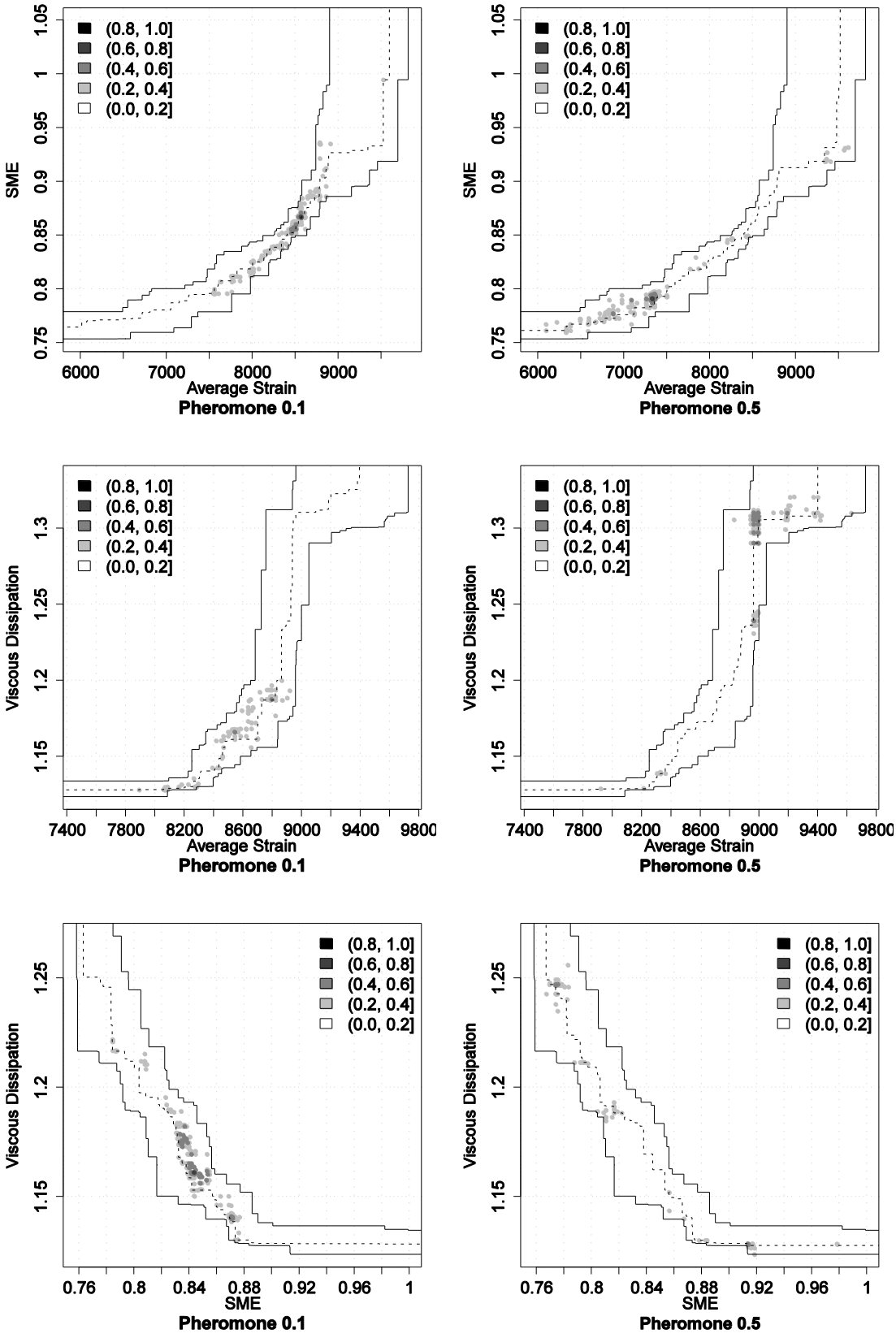

Figure B.9. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.1$ are indicated on the left side; those in favor of $\rho = 0.5$ on the right side.

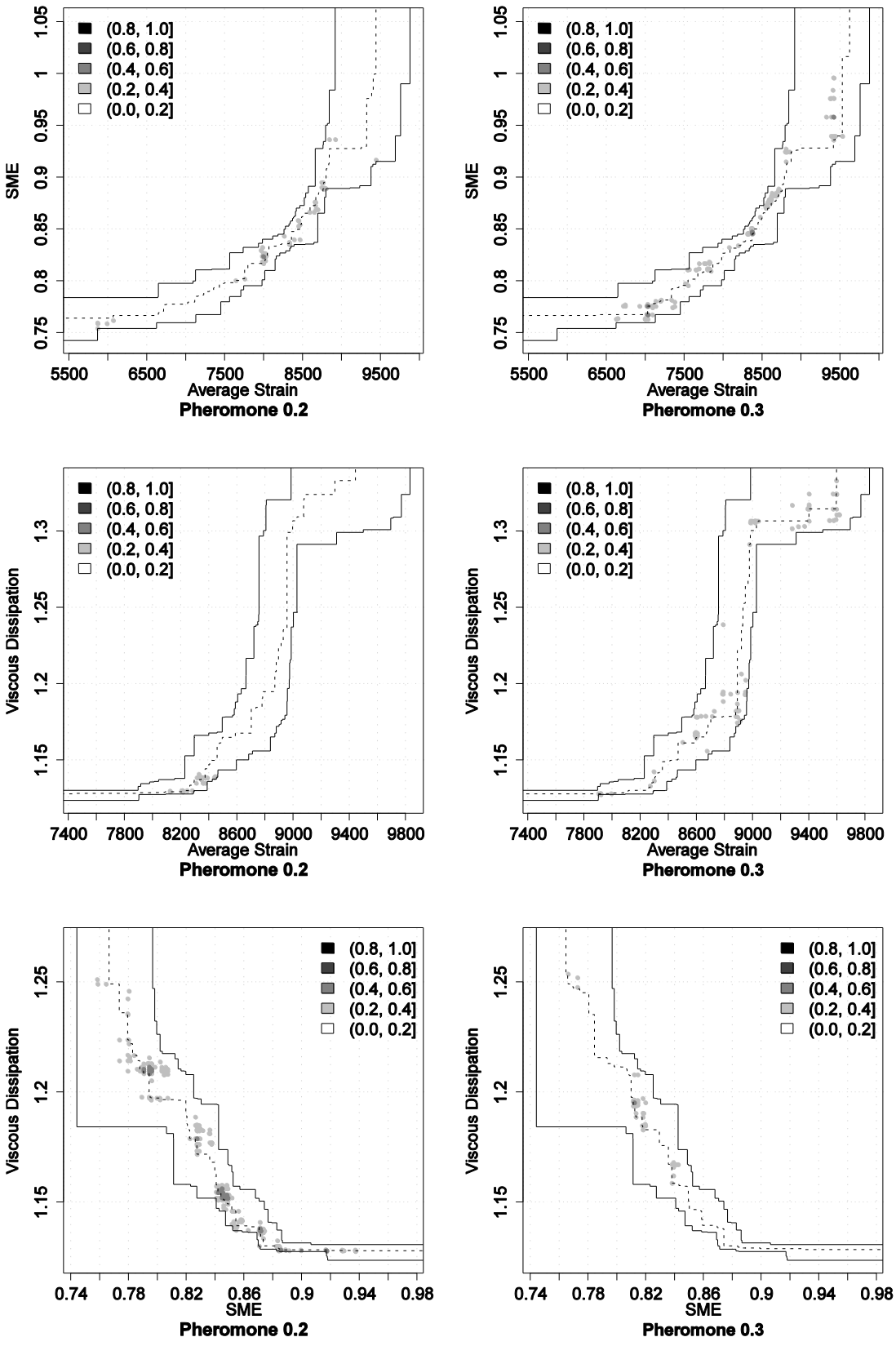

Figure B.10. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.2$ are indicated on the left side; those in favor of $\rho = 0.3$ on the right side.

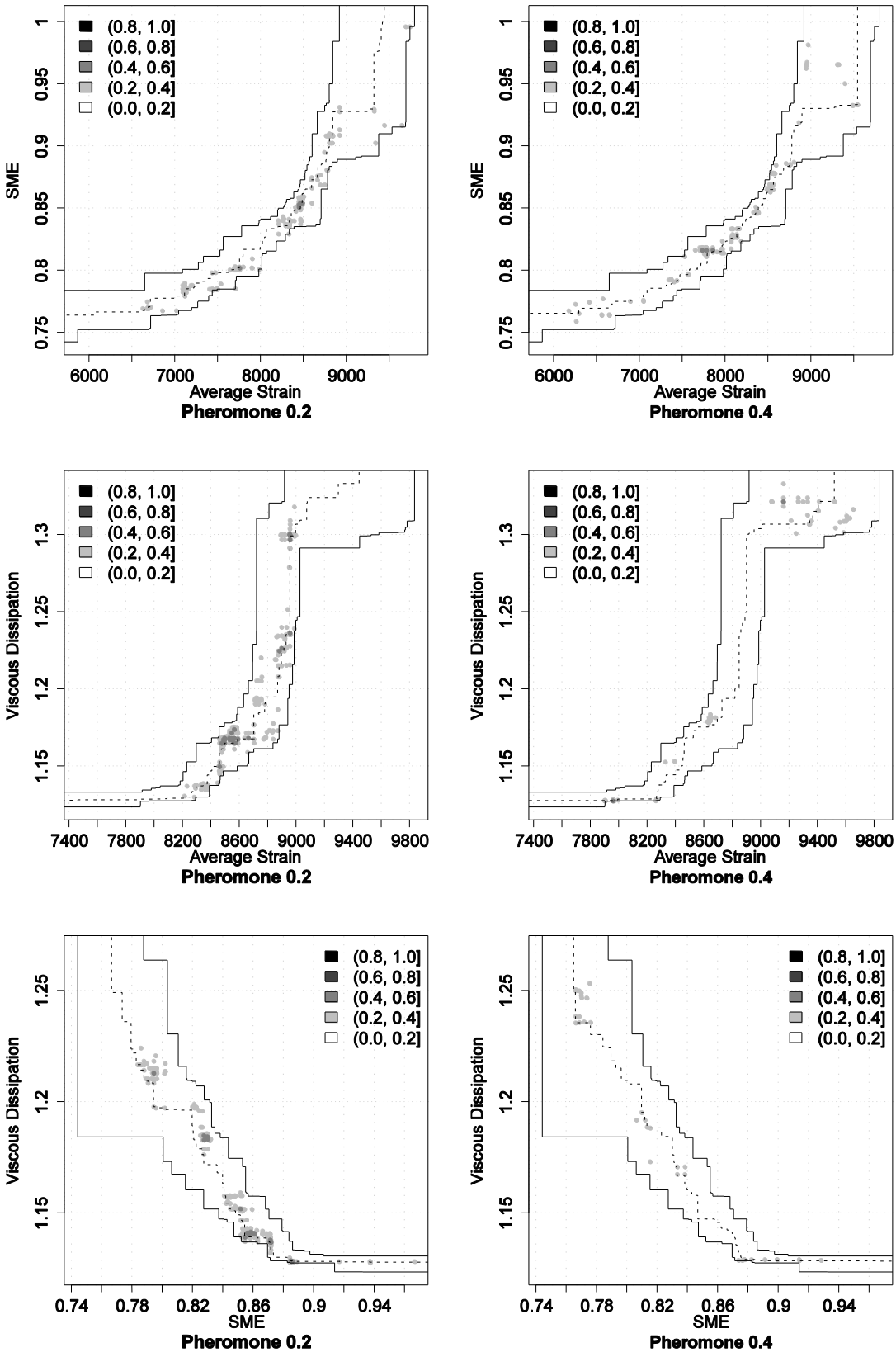

Figure B.11. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.2$ are indicated on the left side; those in favor of $\rho = 0.4$ on the right side.

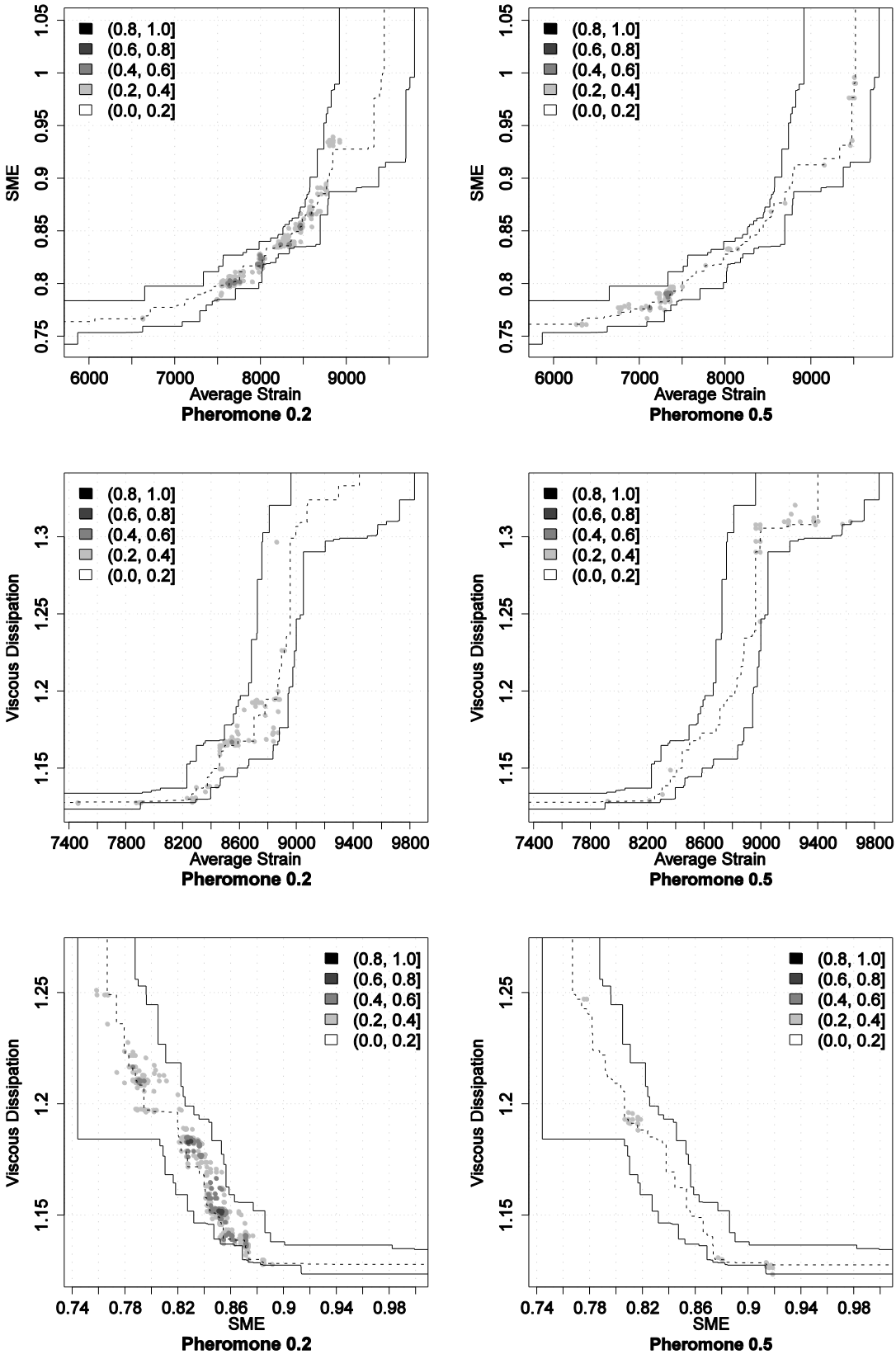

Figure B.12. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.2$ are indicated on the left side; those in favor of $\rho = 0.5$ on the right side.

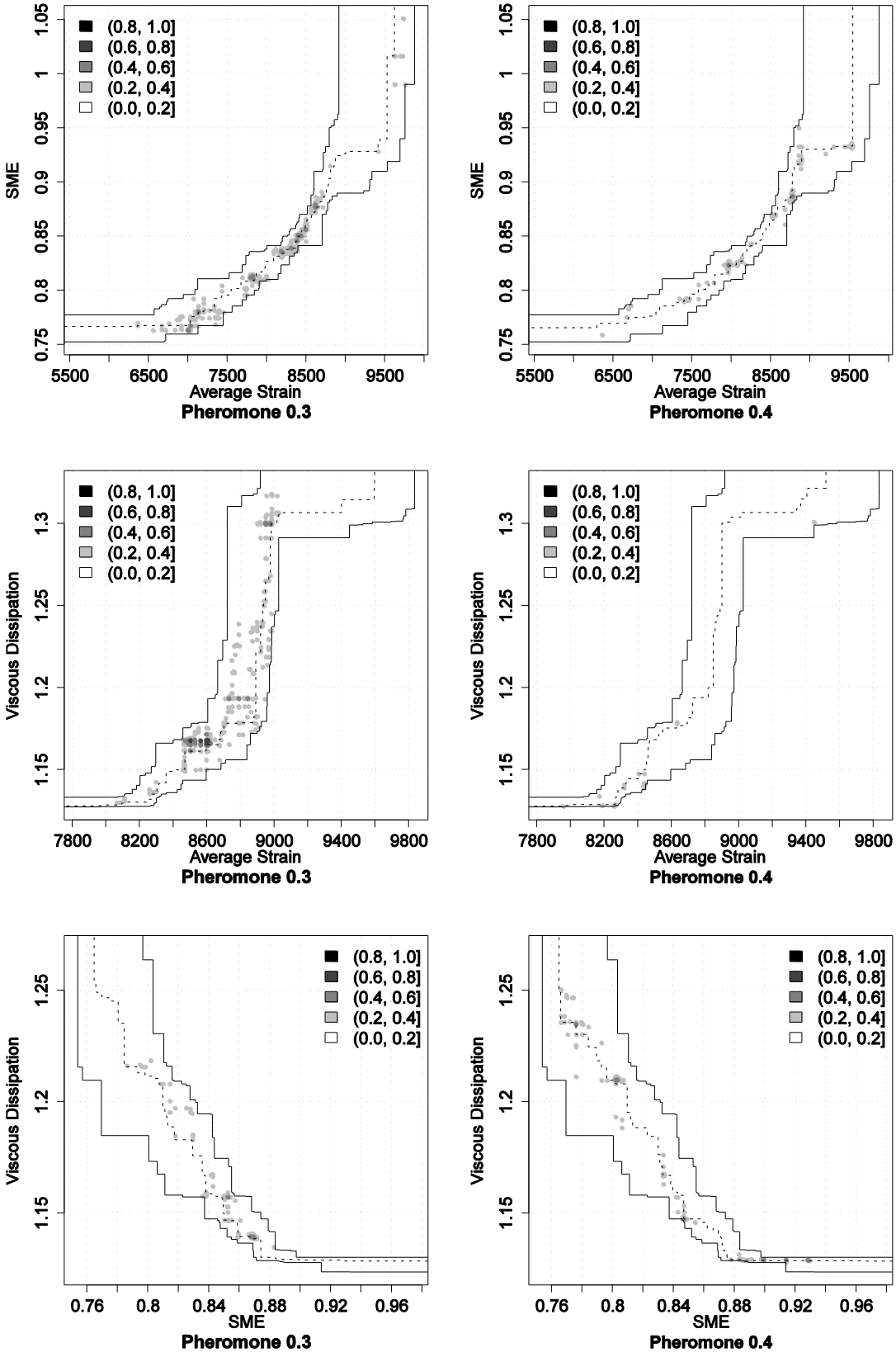

Figure B.13. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.3$ are indicated on the left side; those in favor of $\rho = 0.4$ on the right side.

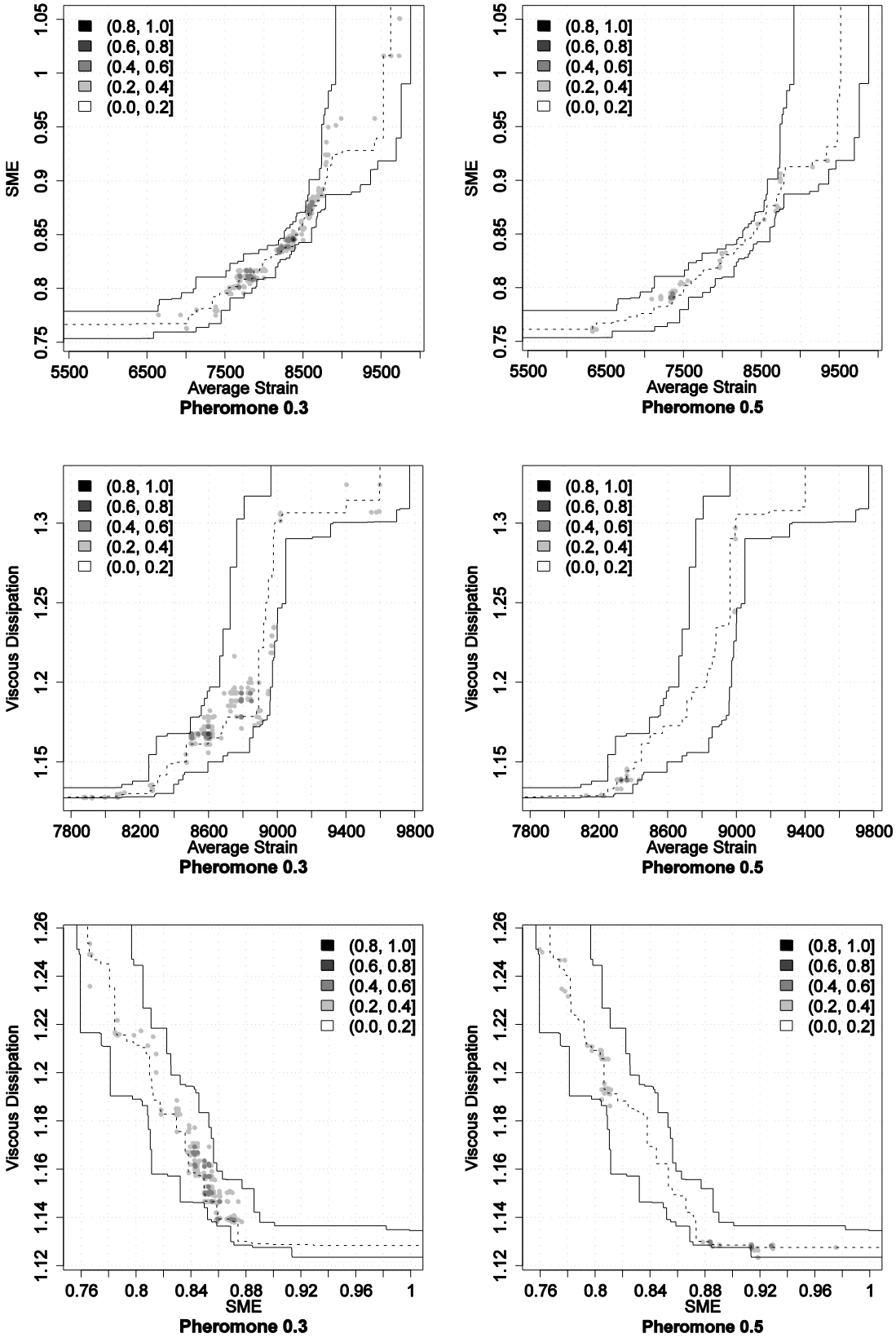

Figure B.14. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.3$ are indicated on the left side; those in favor of $\rho = 0.5$ on the right side.

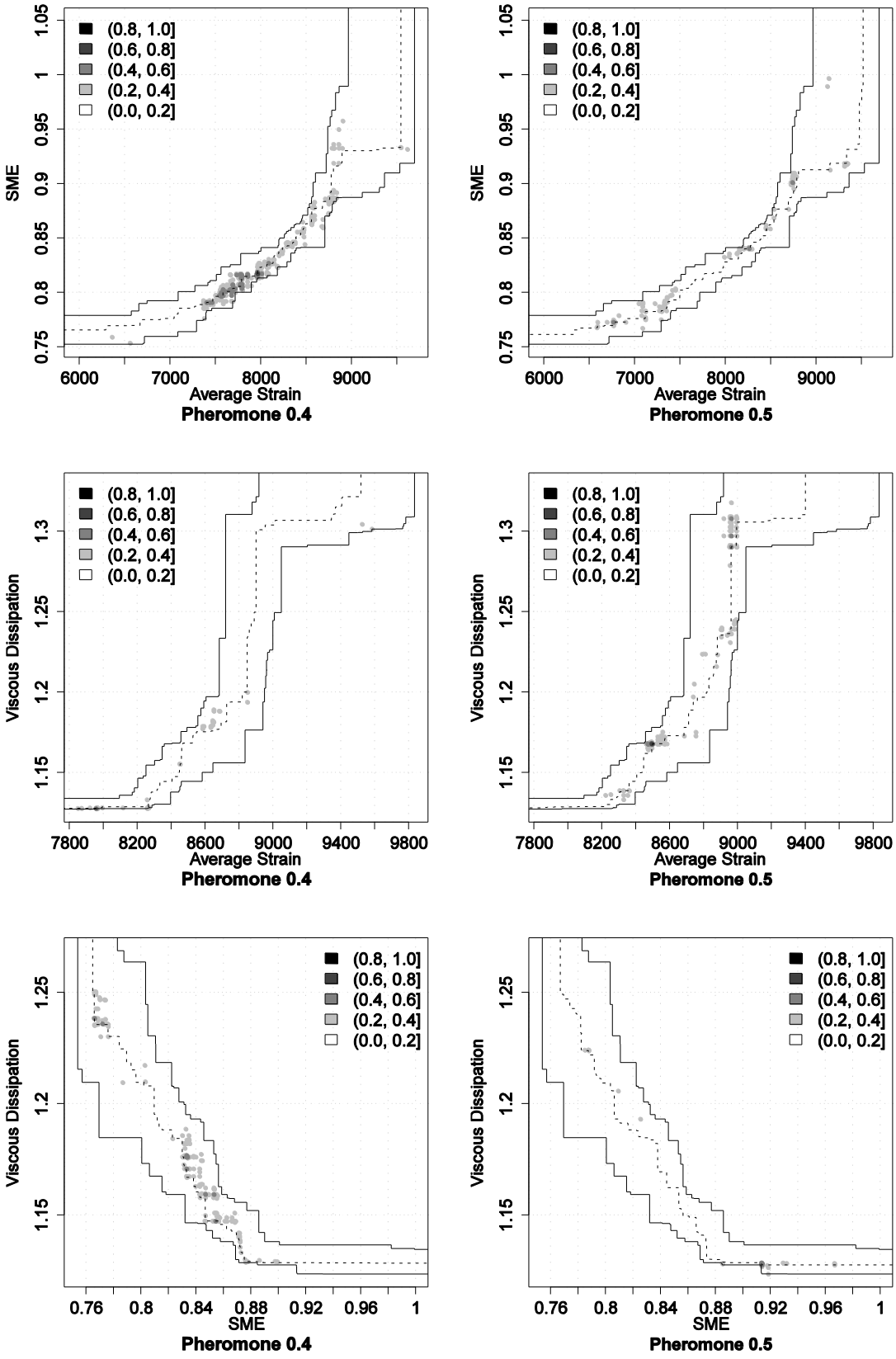

Figure B.15. Influence of the pheromone trail evaporation factor ρ for TSCP4 in terms of empirical attainment functions. Advantages in favor of $\rho = 0.4$ are indicated on the left side; those in favor of $\rho = 0.5$ on the right side.

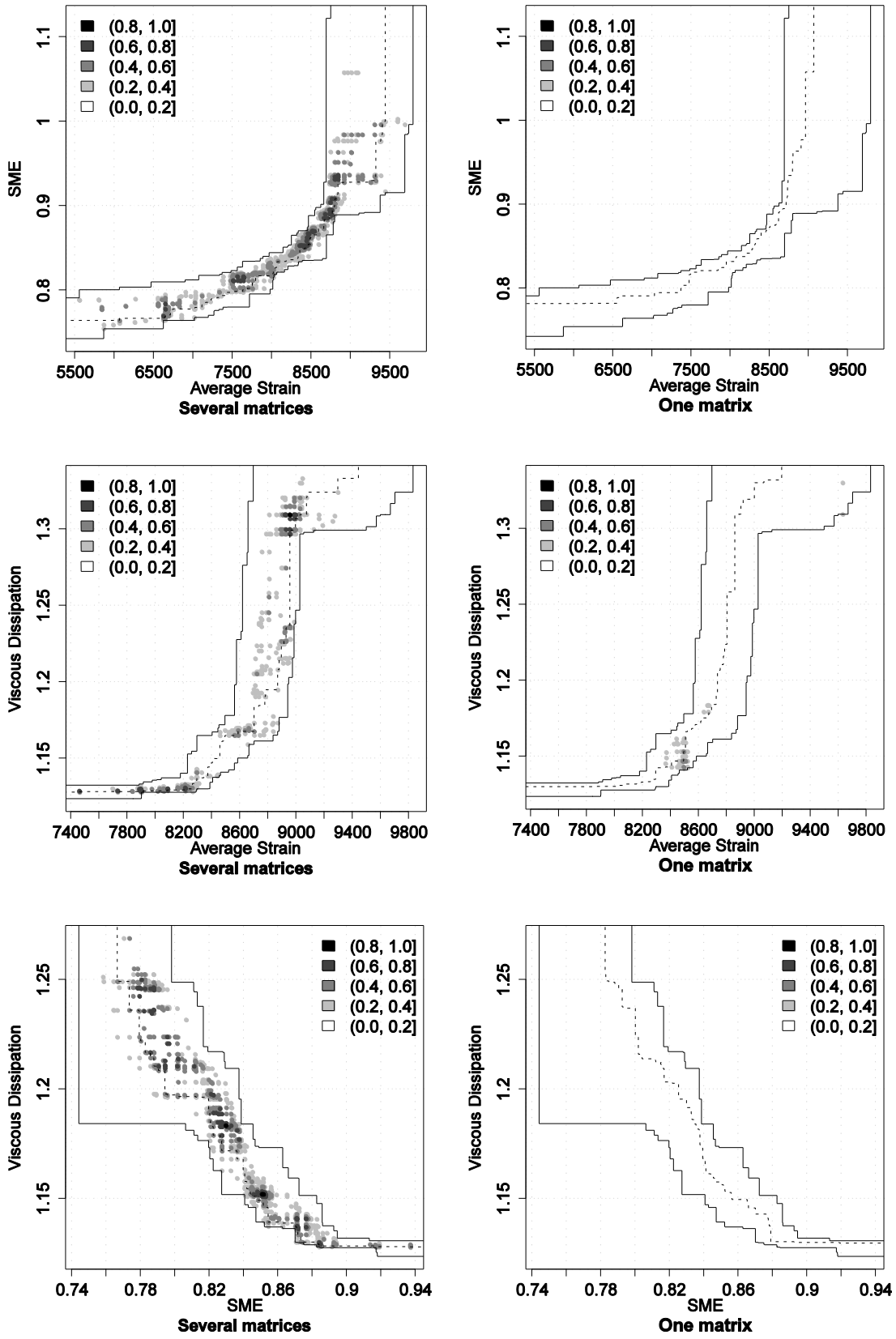

Figure B.16. Influence of the number of pheromone matrices for TSCP4 in terms of empirical attainment functions. Advantages in favor of the utilization of several pheromone matrices (one for each objective) are indicated on the left side; those in favor of the utilization of a single pheromone trail matrix on the right side.

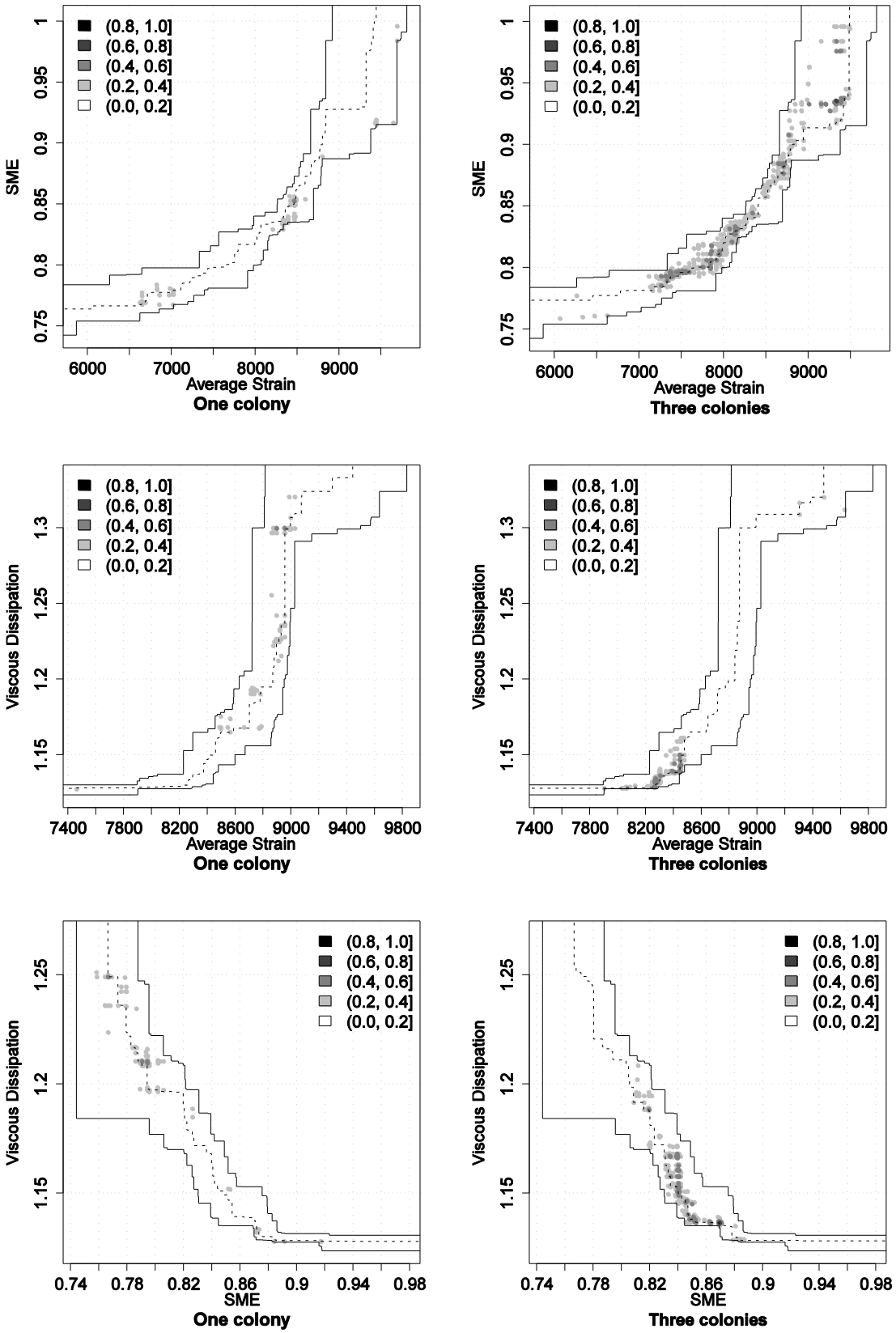

Figure B.17. Influence of the number of colonies for TSCP4 in terms of empirical attainment functions. Advantages in favor of the application of a single colony are indicated on the left side; those in favor of the application of three colonies on the right side.

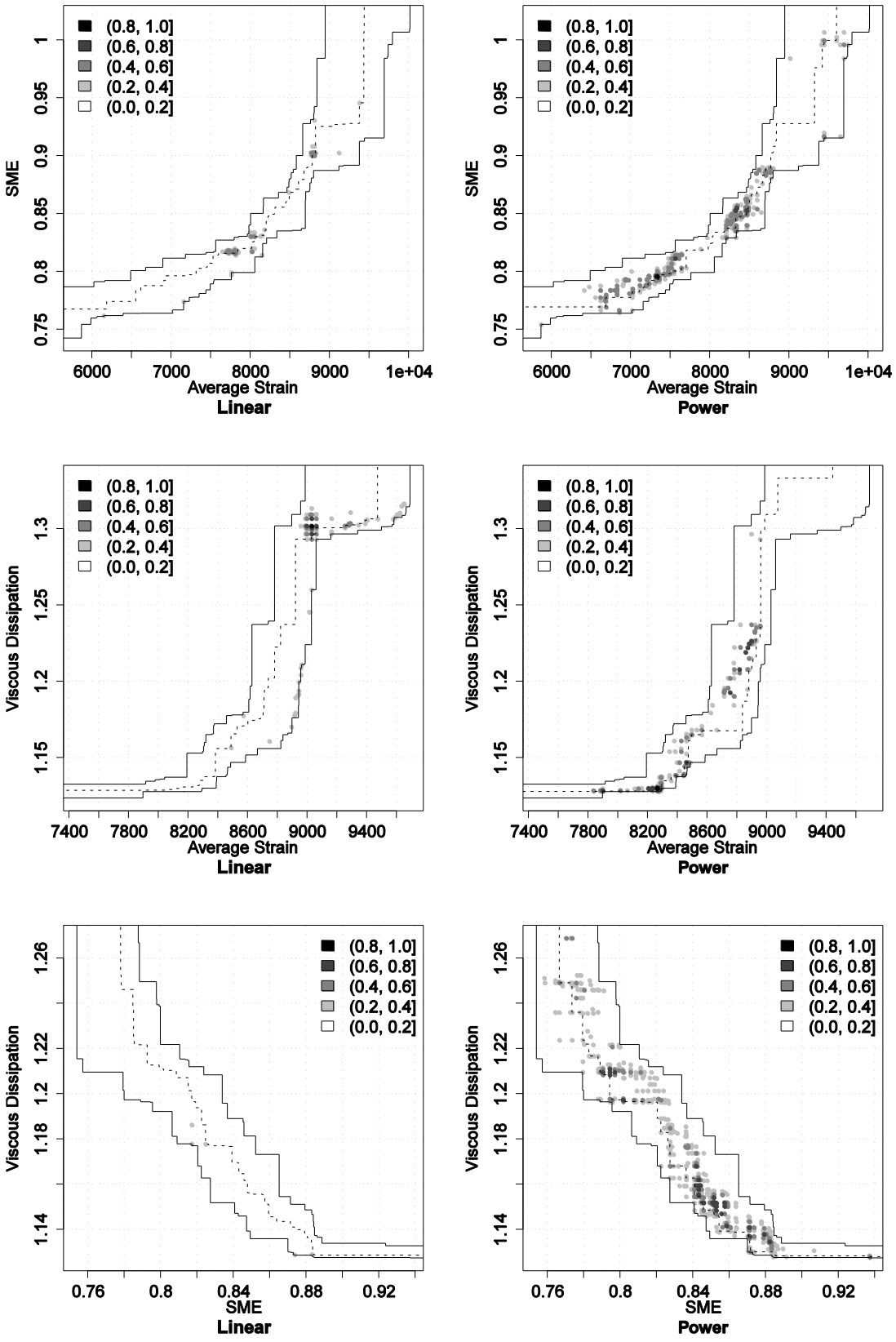

Figure B.18. Influence of the weight aggregation method for TSCP4 in terms of empirical attainment functions. Advantages in favor of the linear aggregation are indicated on the left side; those in favor of the product aggregation on the right side.

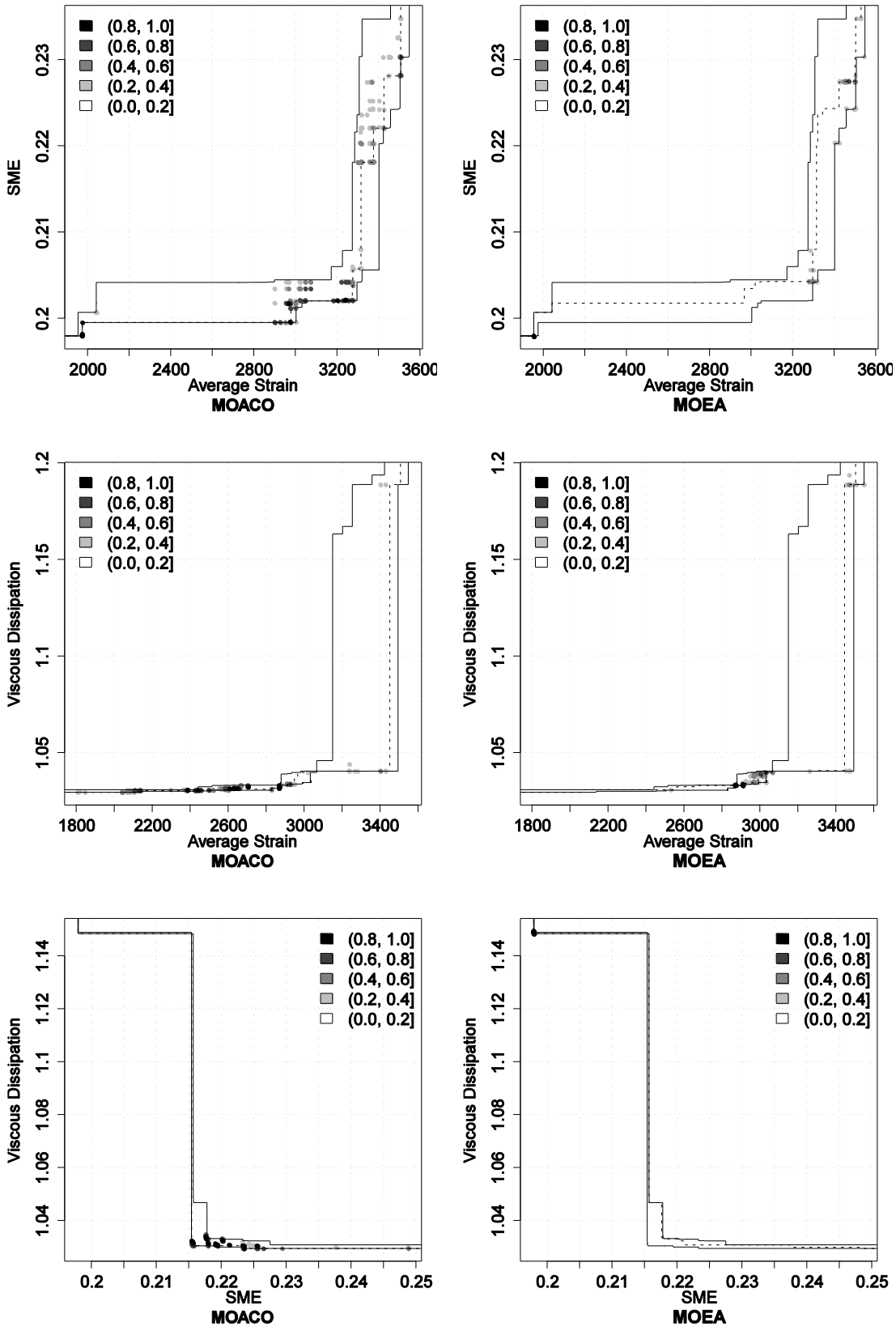

Figure B.19. Results for instance TSCP1. Differences in terms of empirical attainment functions between the multi-objective ant colony optimization algorithm (MOACO) and multi-objective evolutionary algorithm (MOEA) after 3000 evaluations of the simulation program. Advantages in favor of MOACO are indicated on the left side; those in favor of MOEA on the right side.

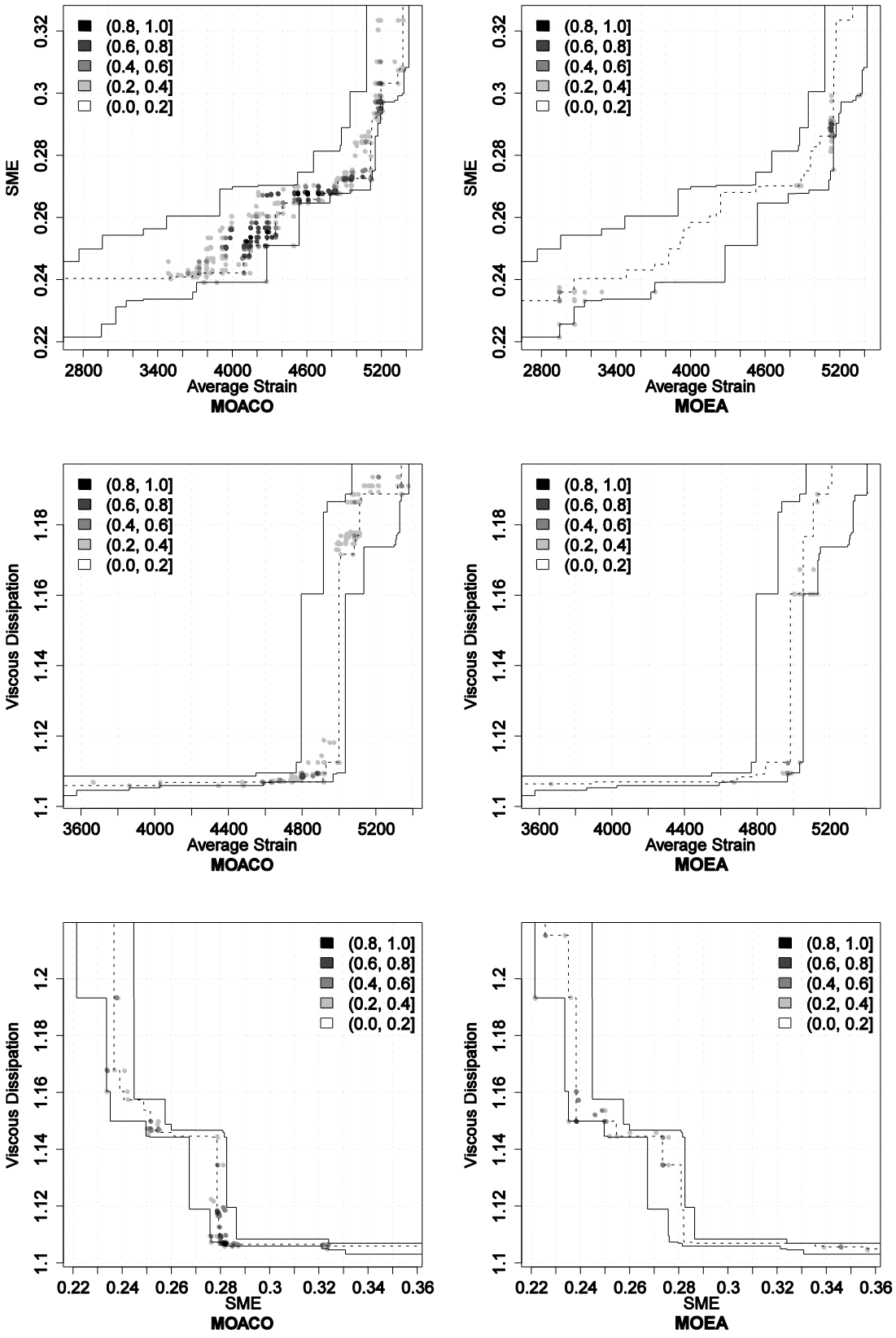

Figure B.20. Results for instance TSCP2. Differences in terms of empirical attainment functions between the multi-objective ant colony optimization algorithm (MOACO) and multi-objective evolutionary algorithm (MOEA) after 3000 evaluations of the simulation program. Advantages in favor of MOACO are indicated on the left side; those in favor of MOEA on the right side.

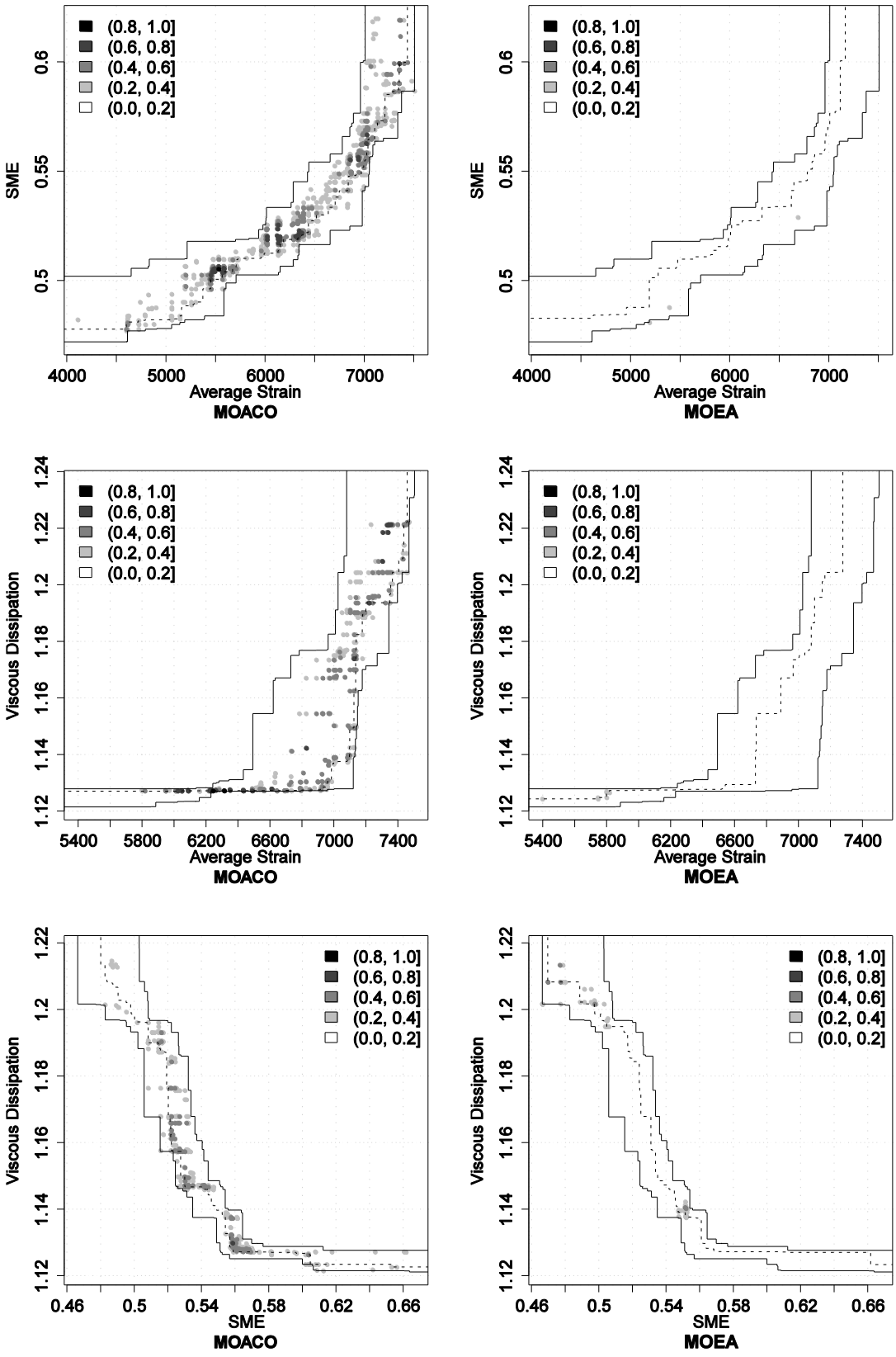

Figure B.21. Results for instance TSCP3. Differences in terms of empirical attainment functions between the multi-objective ant colony optimization algorithm (MOACO) and multi-objective evolutionary algorithm (MOEA) after 3000 evaluations of the simulation program. Advantages in favor of MOACO are indicated on the left side; those in favor of MOEA on the right side.

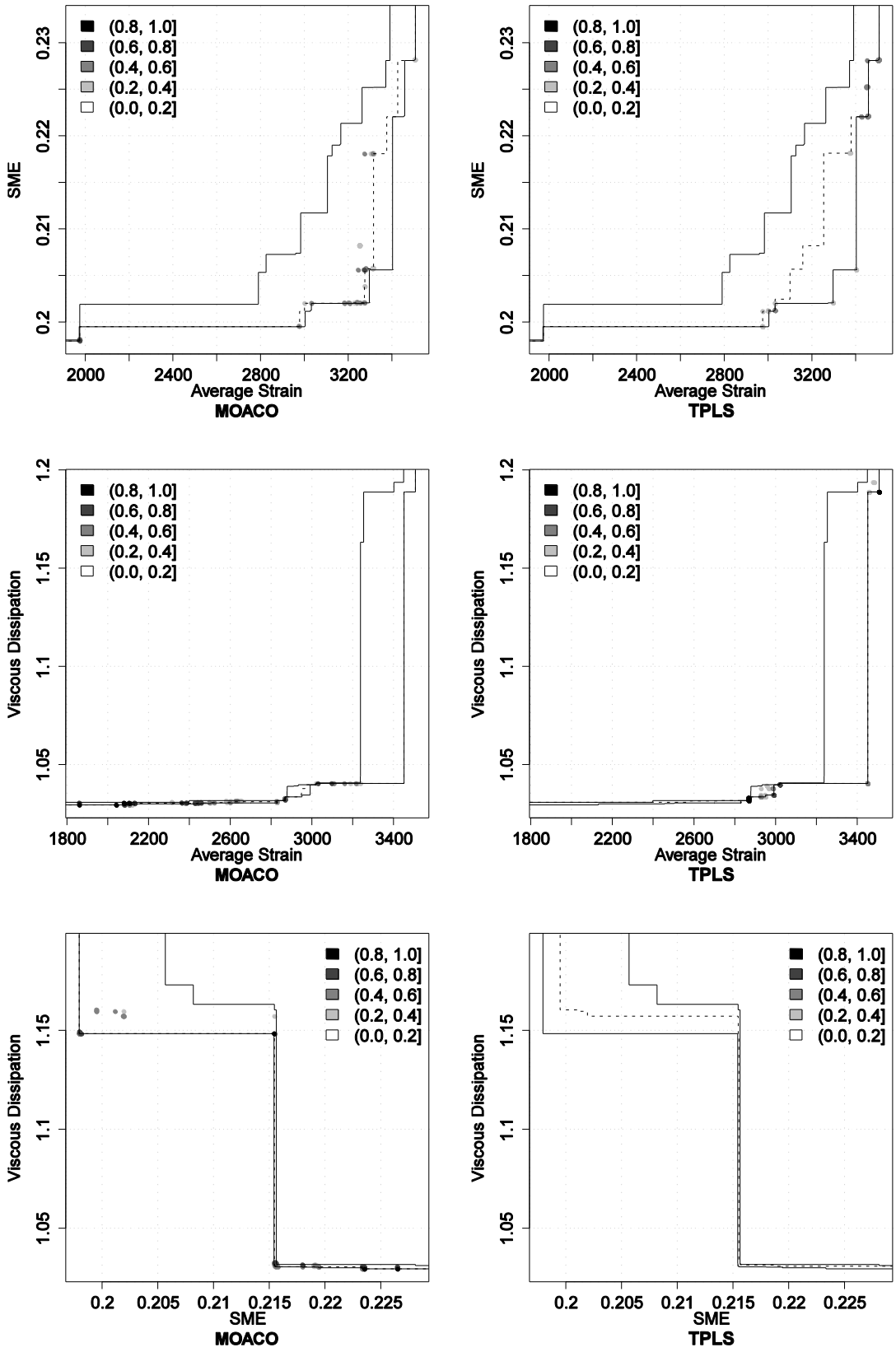

Figure B.22. Results for instance TSCP1. Differences in terms of empirical attainment functions between the multi-objective ant colony optimization algorithm (MOACO) and two-phase local search algorithm (TPLS) after 3000 evaluations of the simulation program. Advantages in favor of MOACO are indicated on the left side; those in favor of TPLS on the right side.

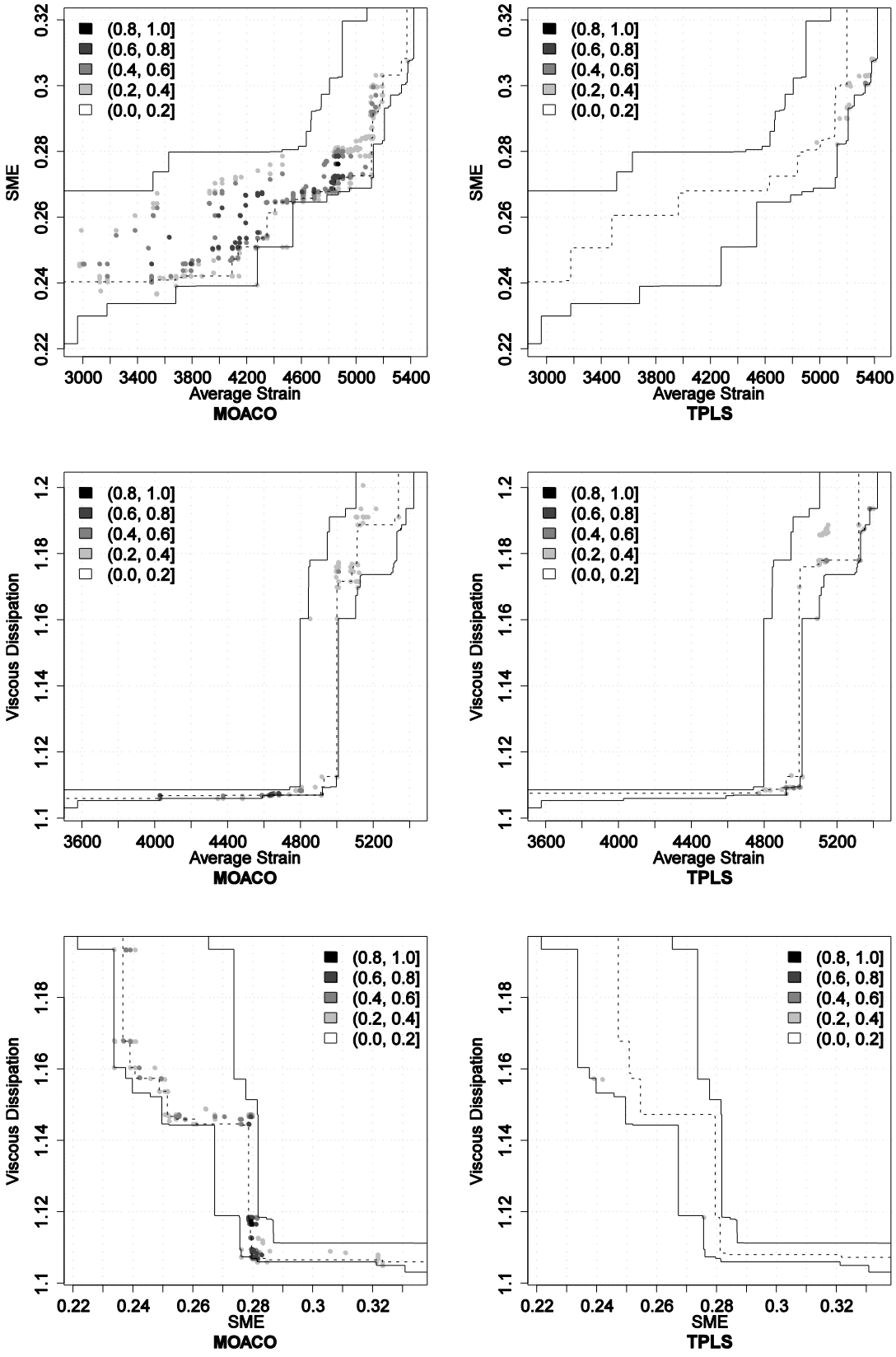

Figure B.23. Results for instance TSCP2. Differences in terms of empirical attainment functions between the multi-objective ant colony optimization algorithm (MOACO) and two-phase local search algorithm (TPLS) after 3000 evaluations of the simulation program. Advantages in favor of MOACO are indicated on the left side; those in favor of TPLS on the right side.

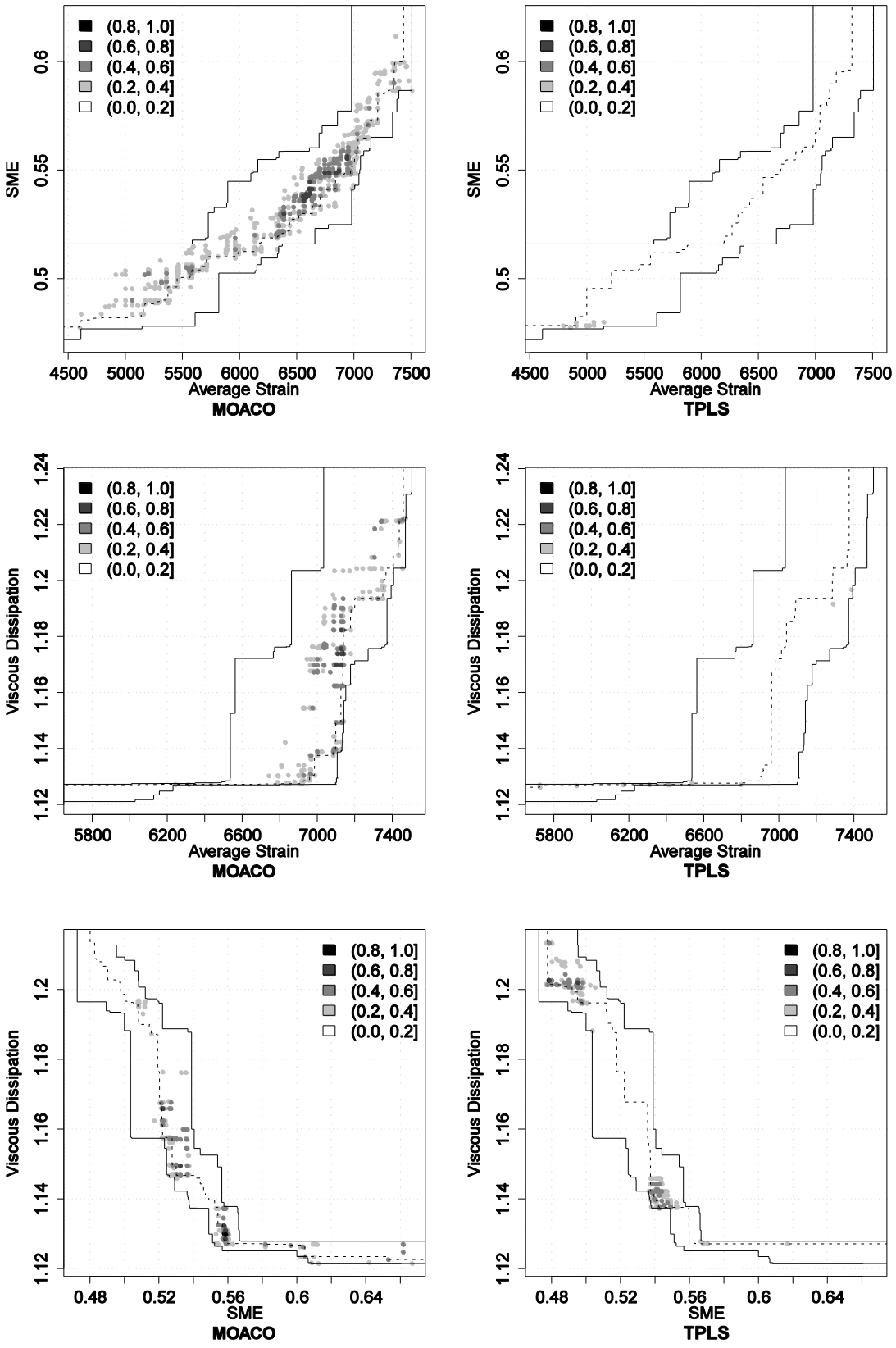

Figure B.24. Results for instance TSCP3. Differences in terms of empirical attainment functions between the multi-objective ant colony optimization algorithm (MOACO) and two-phase local search algorithm (TPLS) after 3000 evaluations of the simulation program. Advantages in favor of MOACO are indicated on the left side; those in favor of TPLS on the right side.

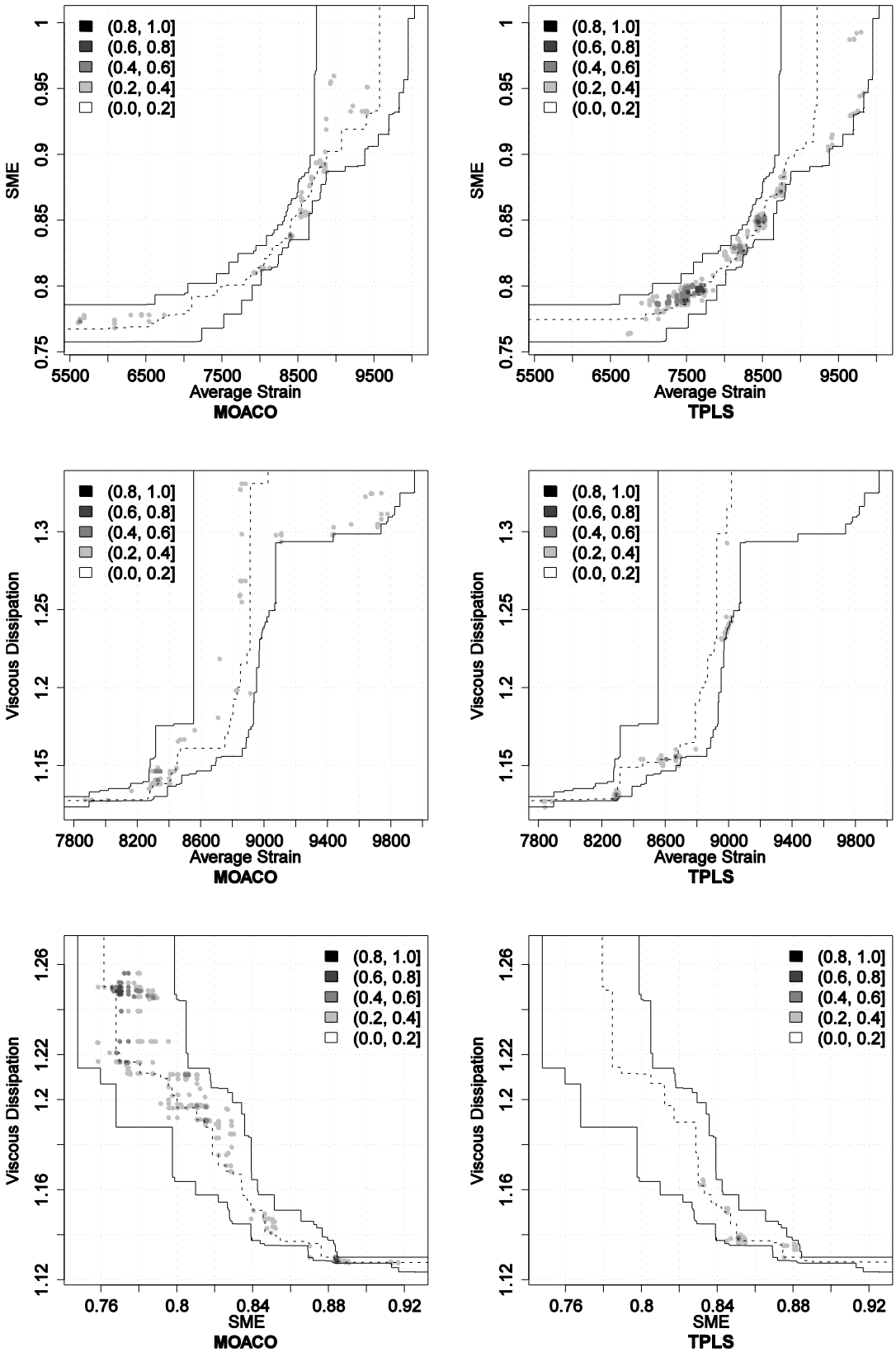

Figure B.25. Results for instance TSCP4. Differences in terms of empirical attainment functions between the multi-objective ant colony optimization algorithm (MOACO) and two-phase local search algorithm (TPLS) after 3000 evaluations of the simulation program. Advantages in favor of MOACO are indicated on the left side; those in favor of TPLS on the right side.

Appendix C

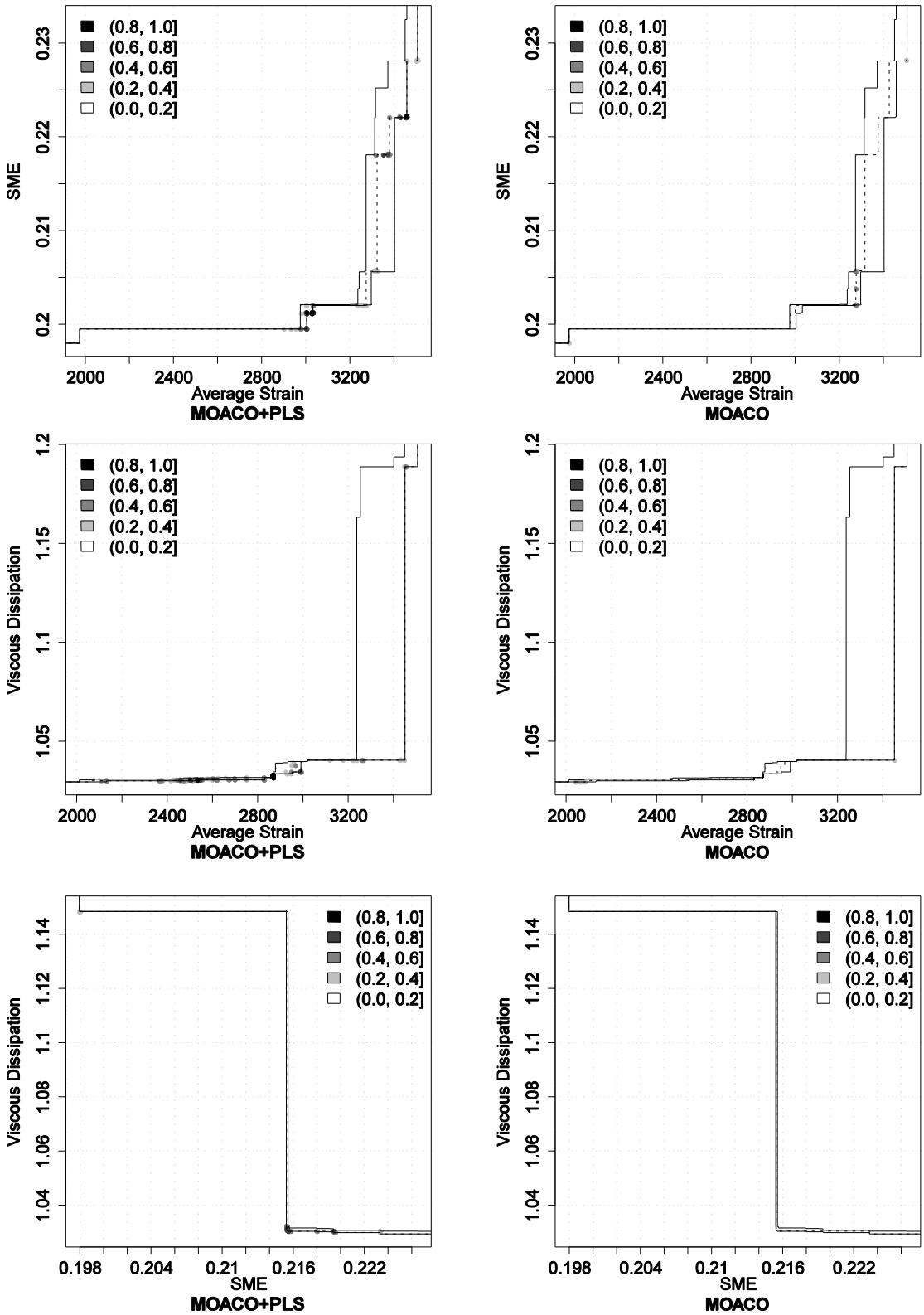

Figure C.1. Results for instance TSCP1. Differences in terms of empirical attainment functions between the MOACO+PLS algorithm and MOACO after 3000 evaluations of the simulation program. Advantages in favor of MOACO+PLS are indicated on the left side; those in favor of MOACO on the right side.

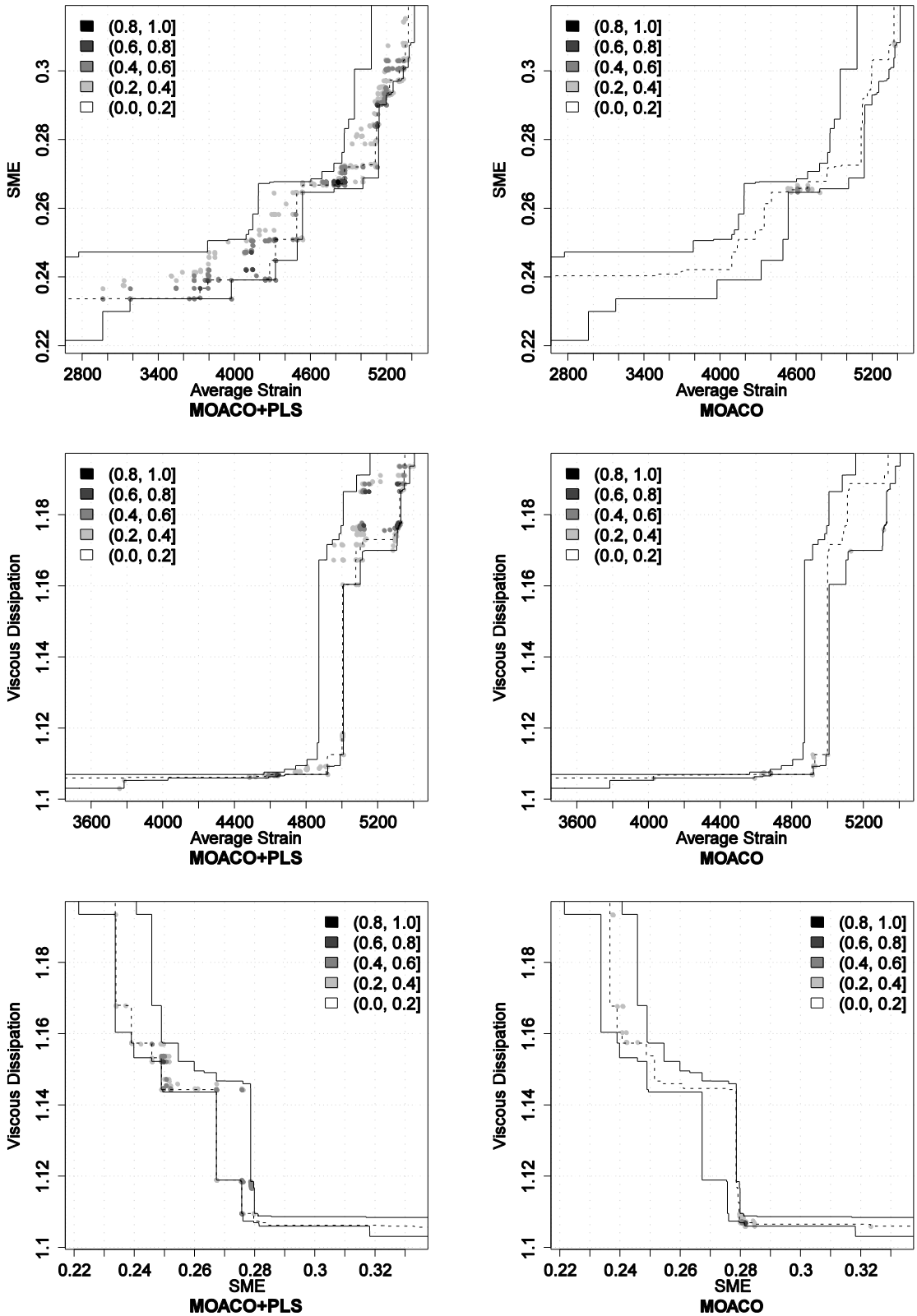

Figure C.2. Results for instance TSCP2 (see the caption of Figure C.1 for more details).

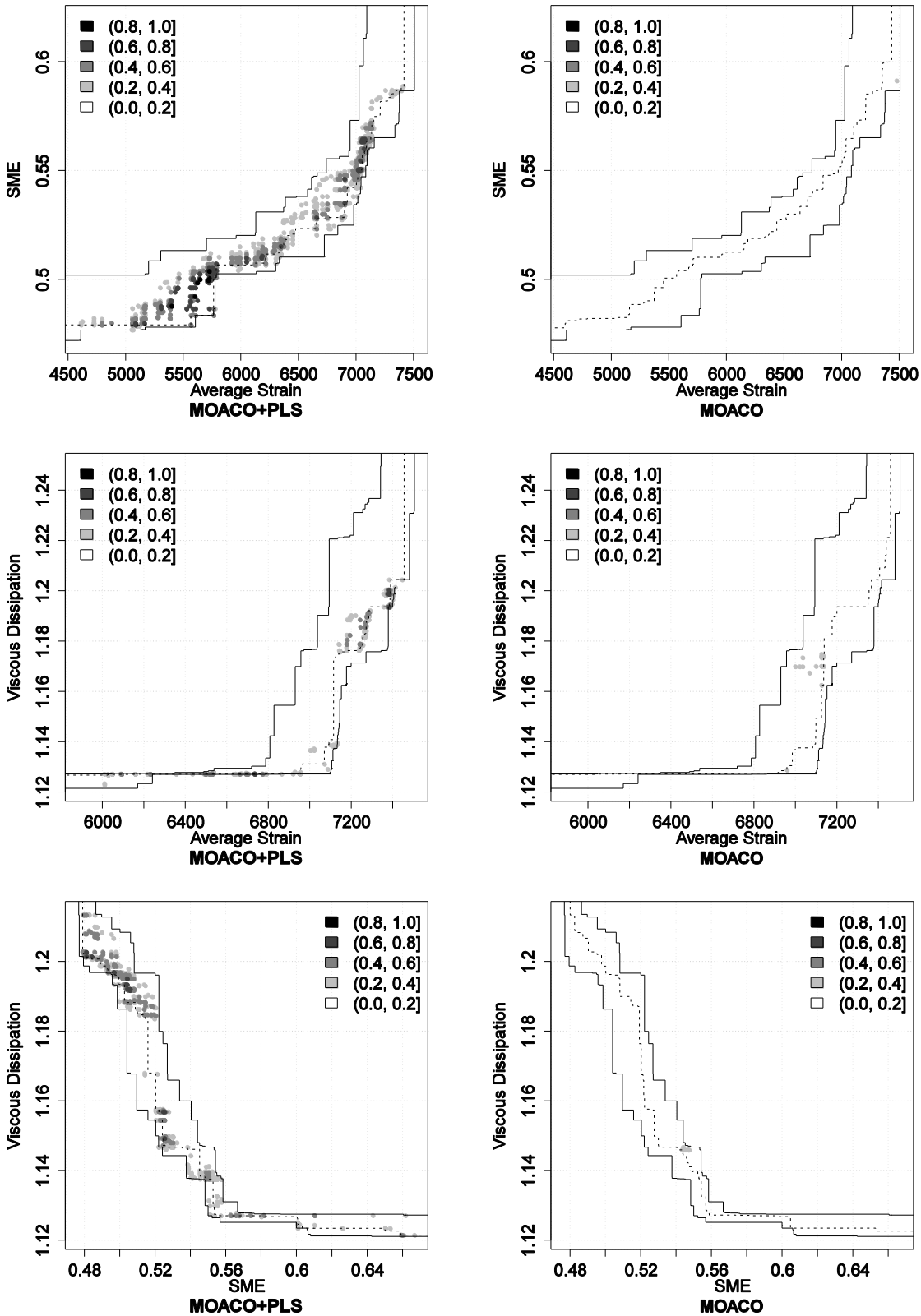

Figure C.3. Results for instance TSCP3 (see the caption of Figure C.1 for more details).

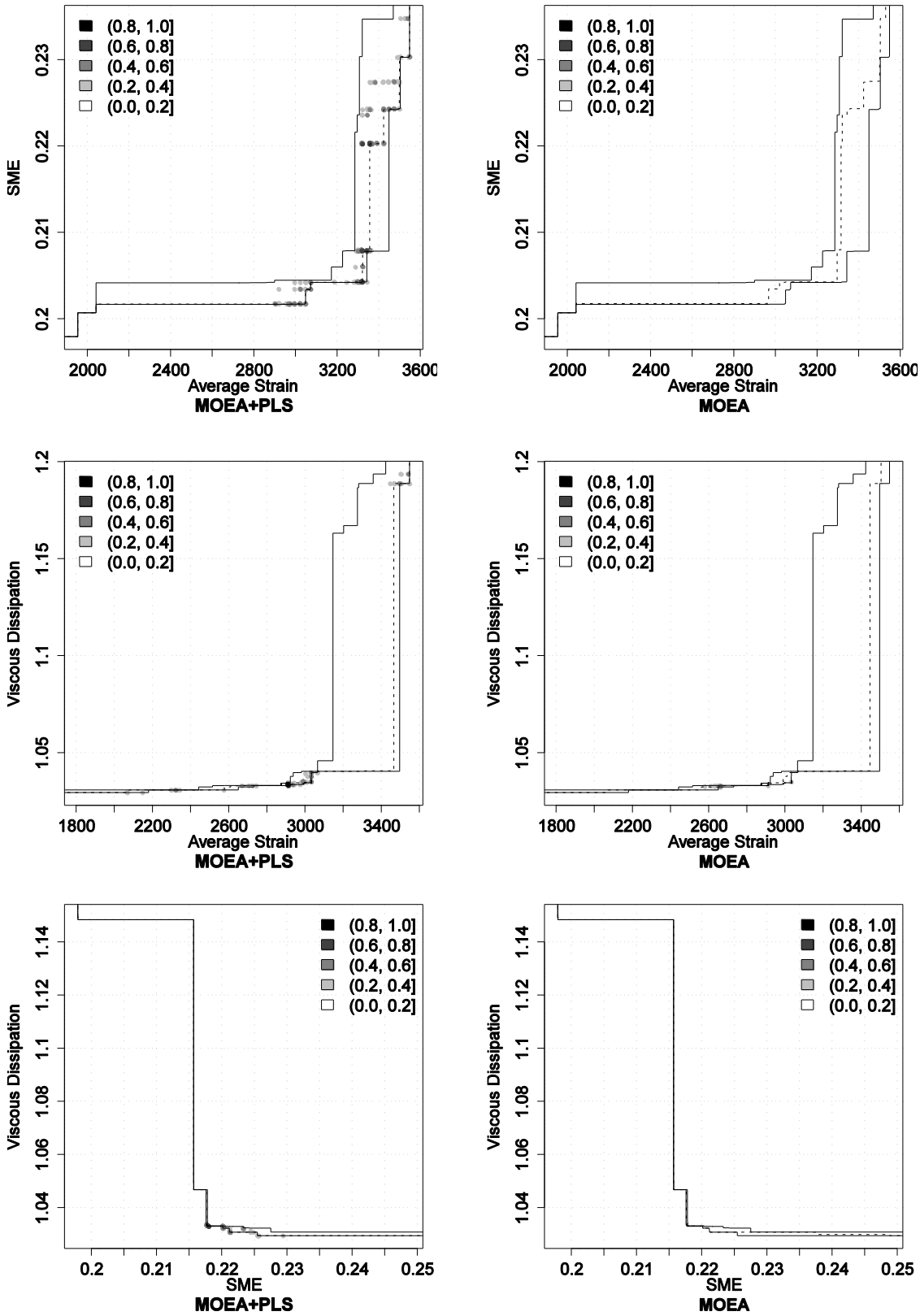

Figure C.4. Results for instance TSCP1. Differences in terms of empirical attainment functions between the MOEA+PLS algorithm and MOEA after 3000 evaluations of the simulation program. Advantages in favor of MOEA+PLS are indicated on the left side; those in favor of MOEA on the right side.

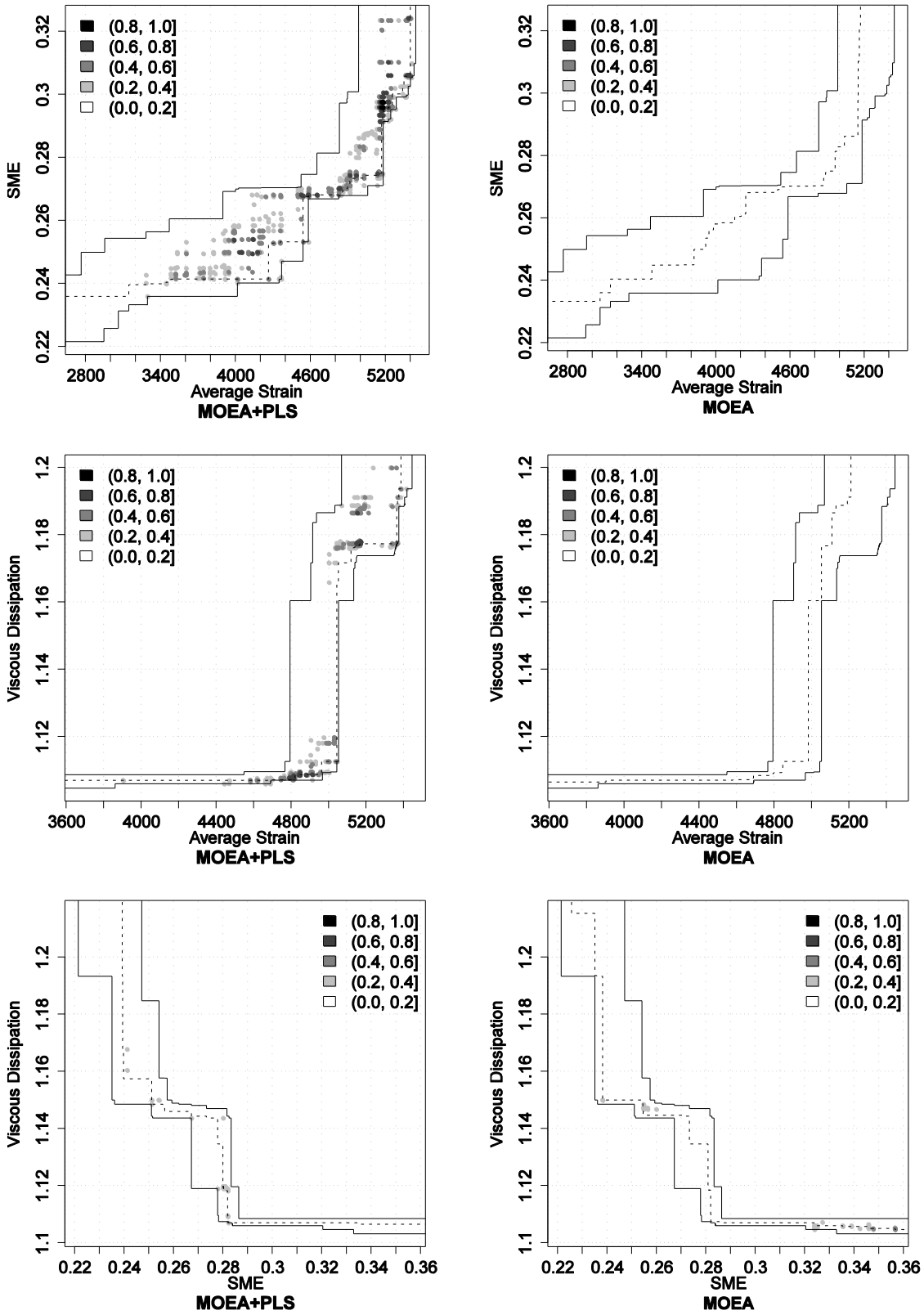

Figure C.5. Results for instance TSCP2 (see the caption of Figure C.4 for more details).

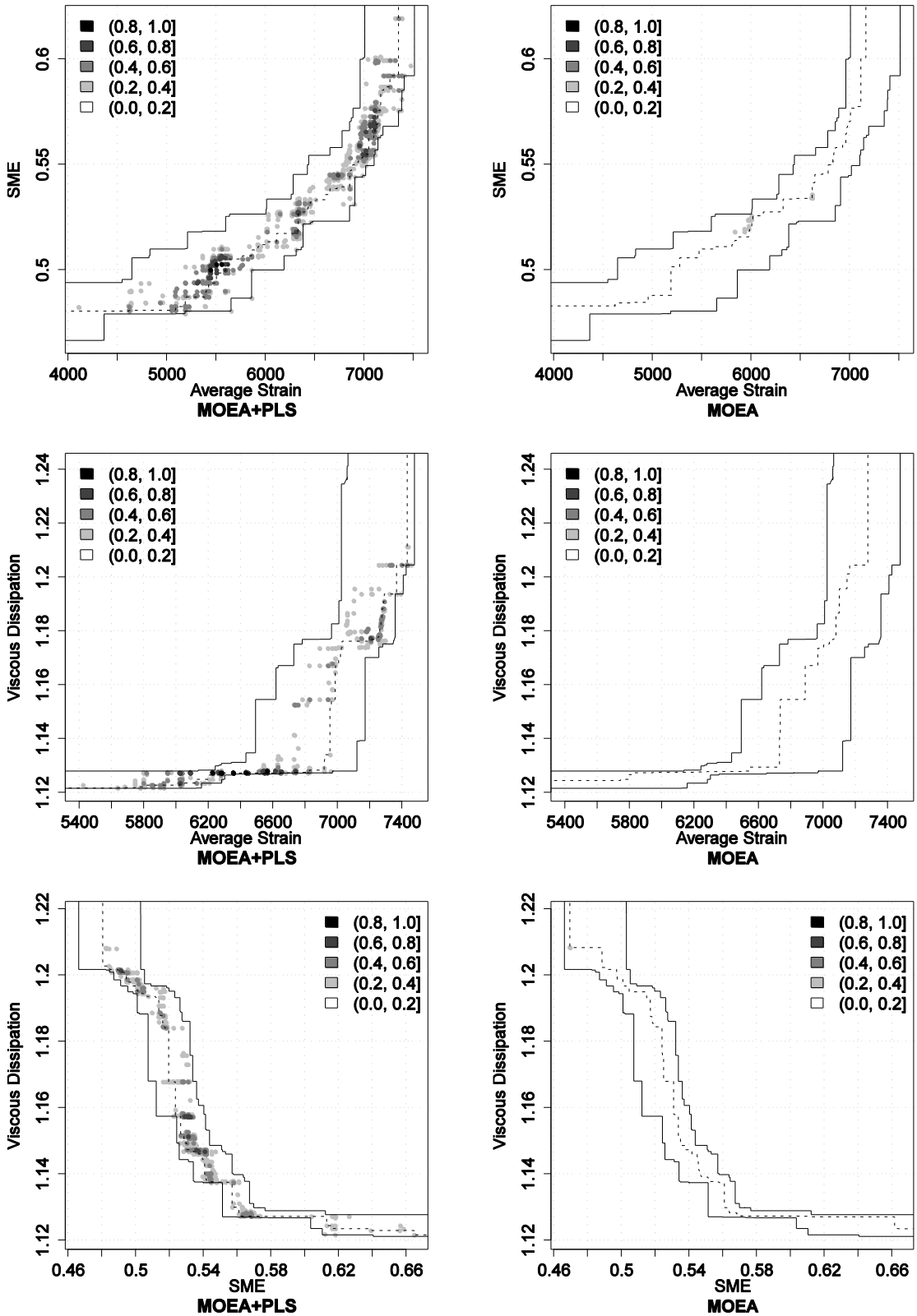

Figure C.6. Results for instance TSCP3 (see the caption of Figure C.4 for more details).

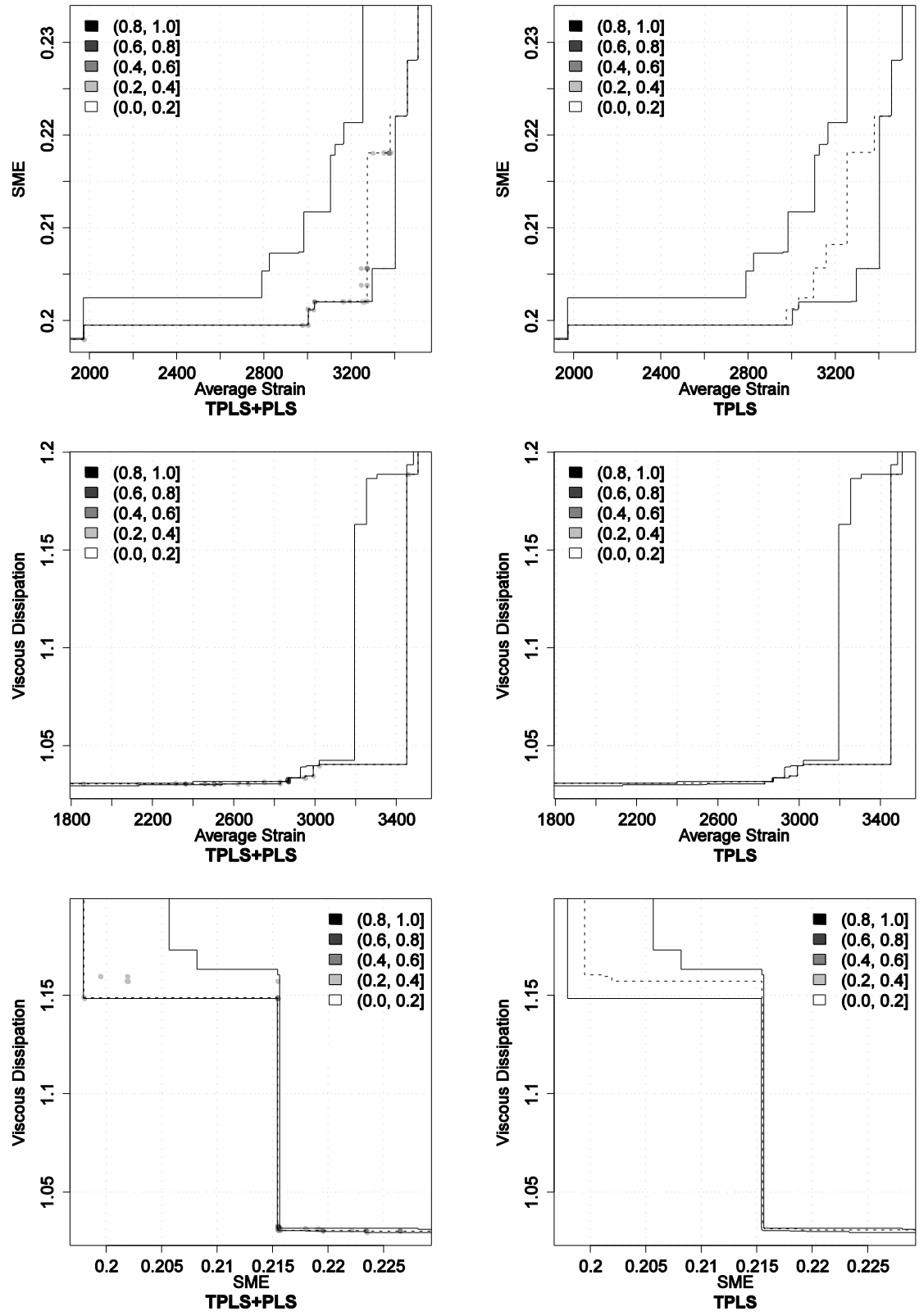

Figure C.7. Results for instance TSCP1. Differences in terms of empirical attainment functions between the TPLS+PLS algorithm and TPLS after 3000 evaluations of the simulation program. Advantages in favor of TPLS+PLS are indicated on the left side; those in favor of TPLS on the right side.

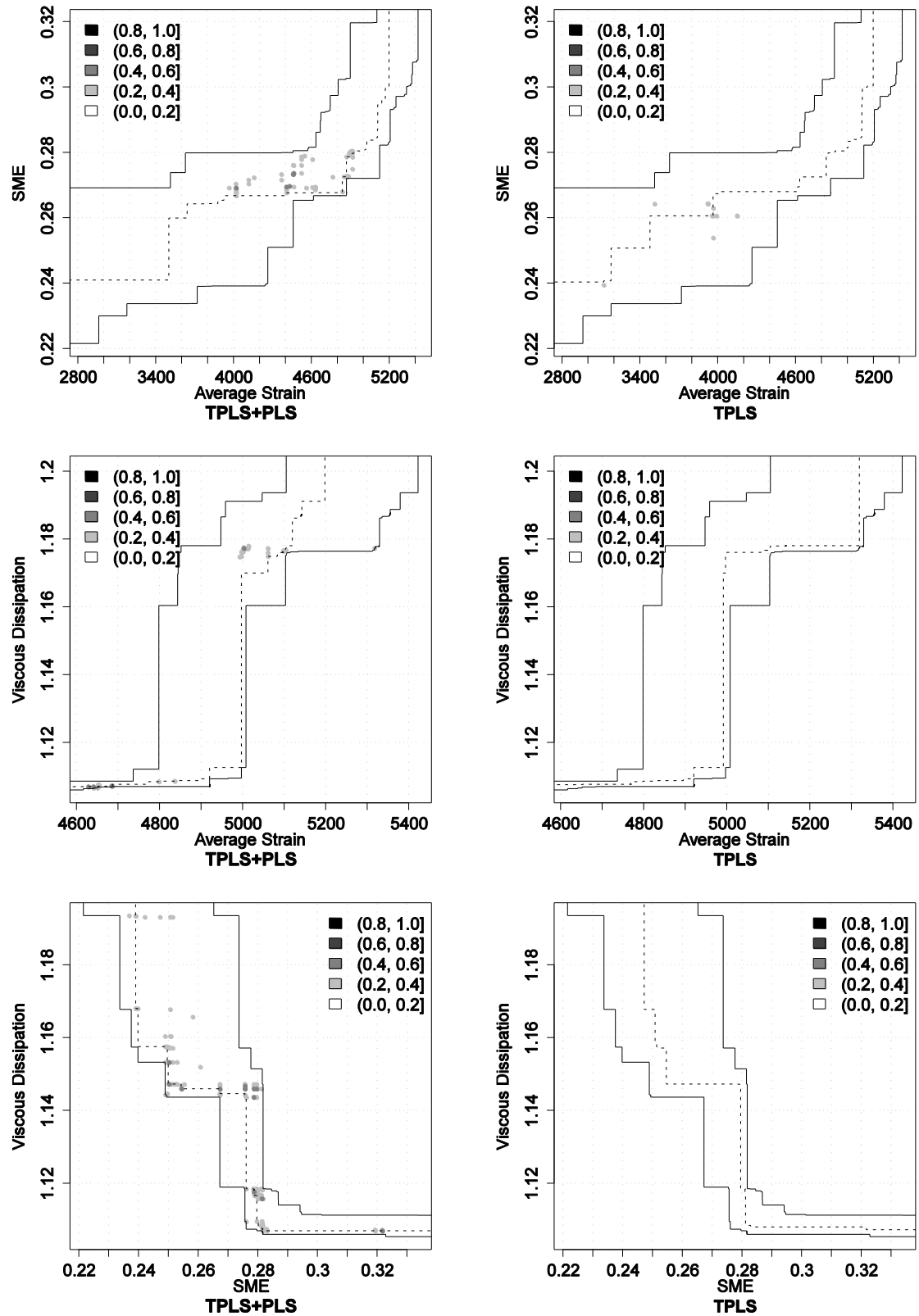

Figure C.8. Results for instance TSCP2 (see the caption of Figure C.7 for more details).

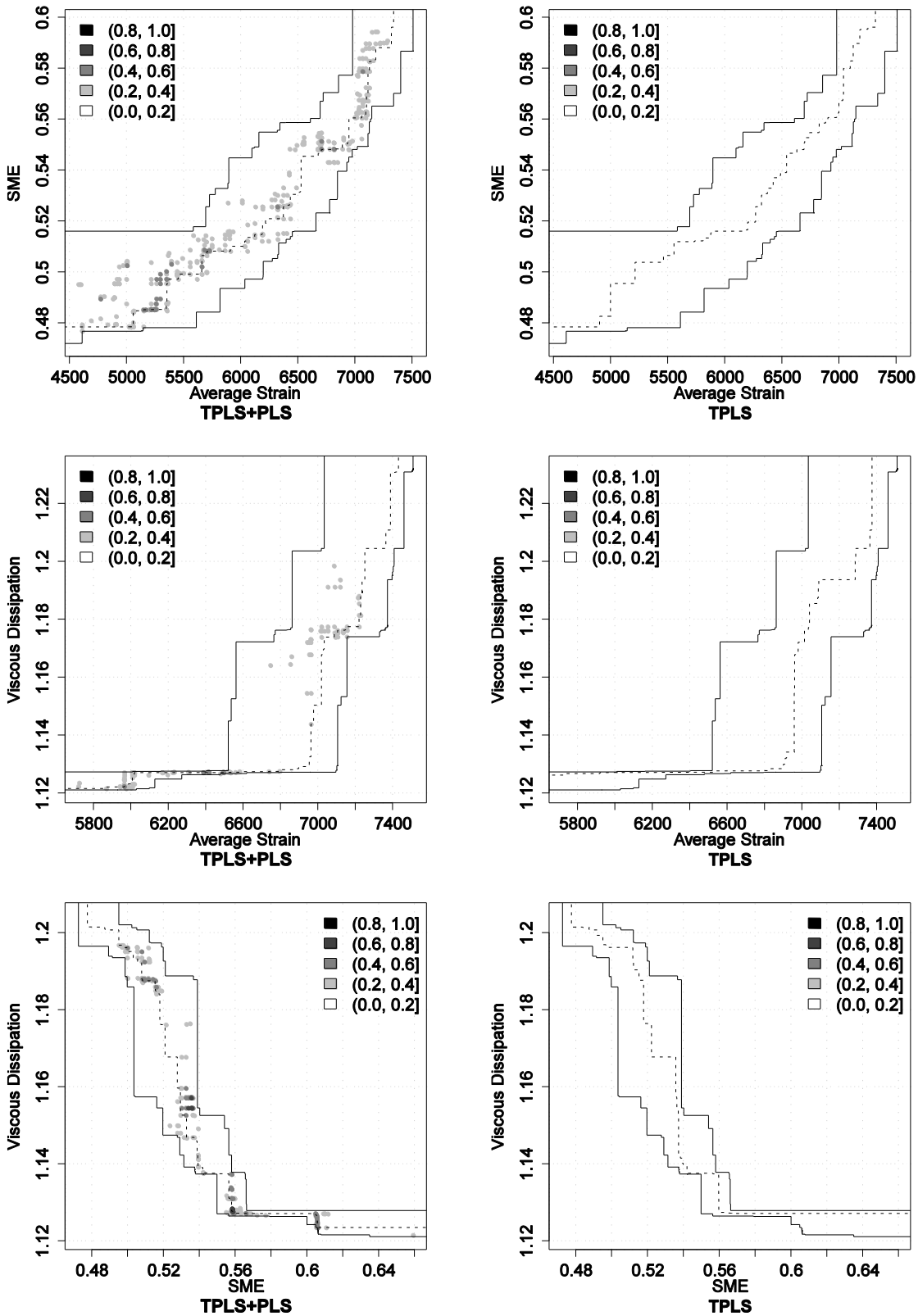

Figure C.9. Results for instance TSCP3 (see the caption of Figure C.7 for more details).

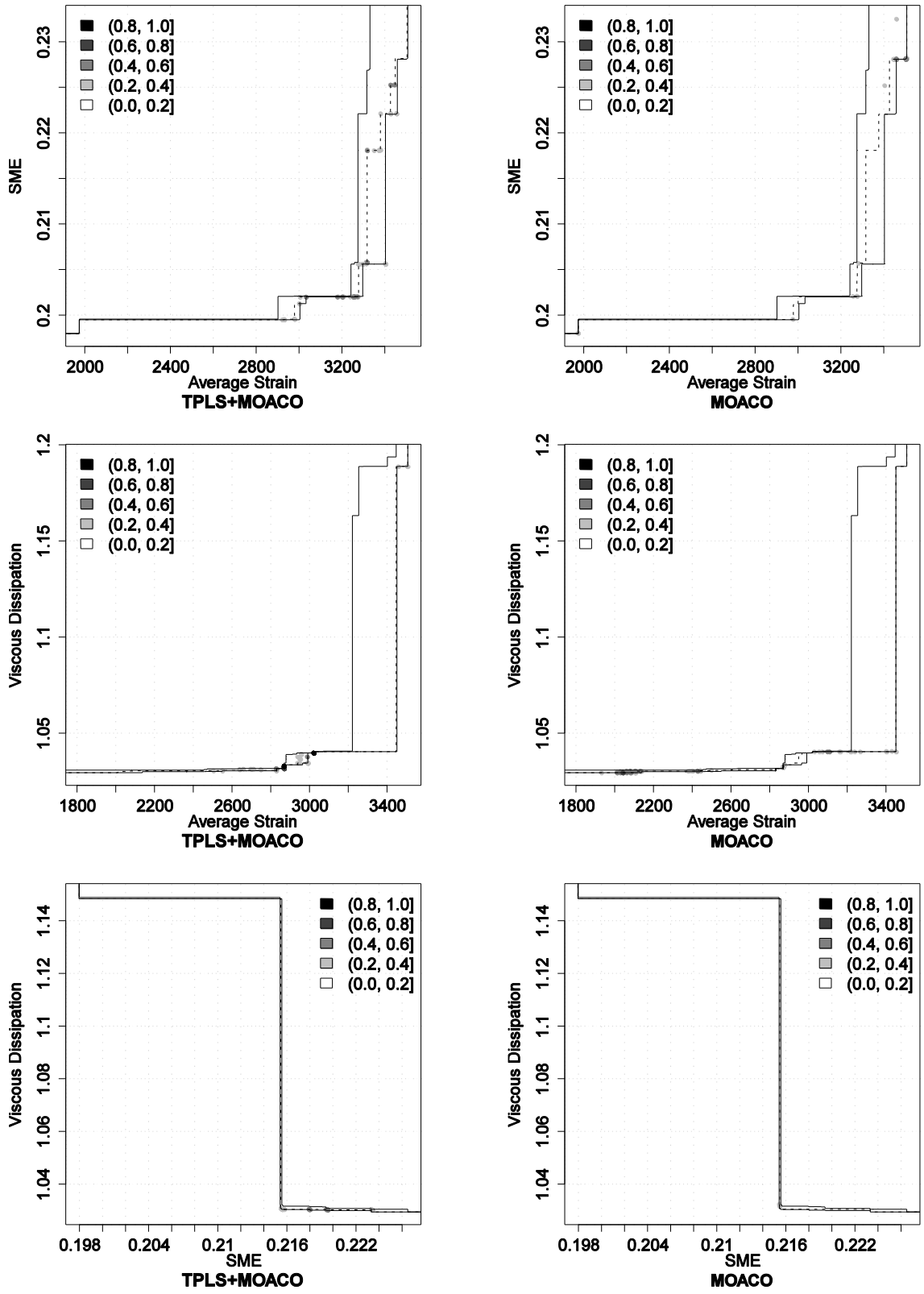

Figure C.10. Results for instance TSCP1. Differences in terms of empirical attainment functions between the TPLS+MOACO algorithm and MOACO after 3000 evaluations of the simulation program. Advantages in favor of TPLS+MOACO are indicated on the left side; those in favor of MOACO on the right side.

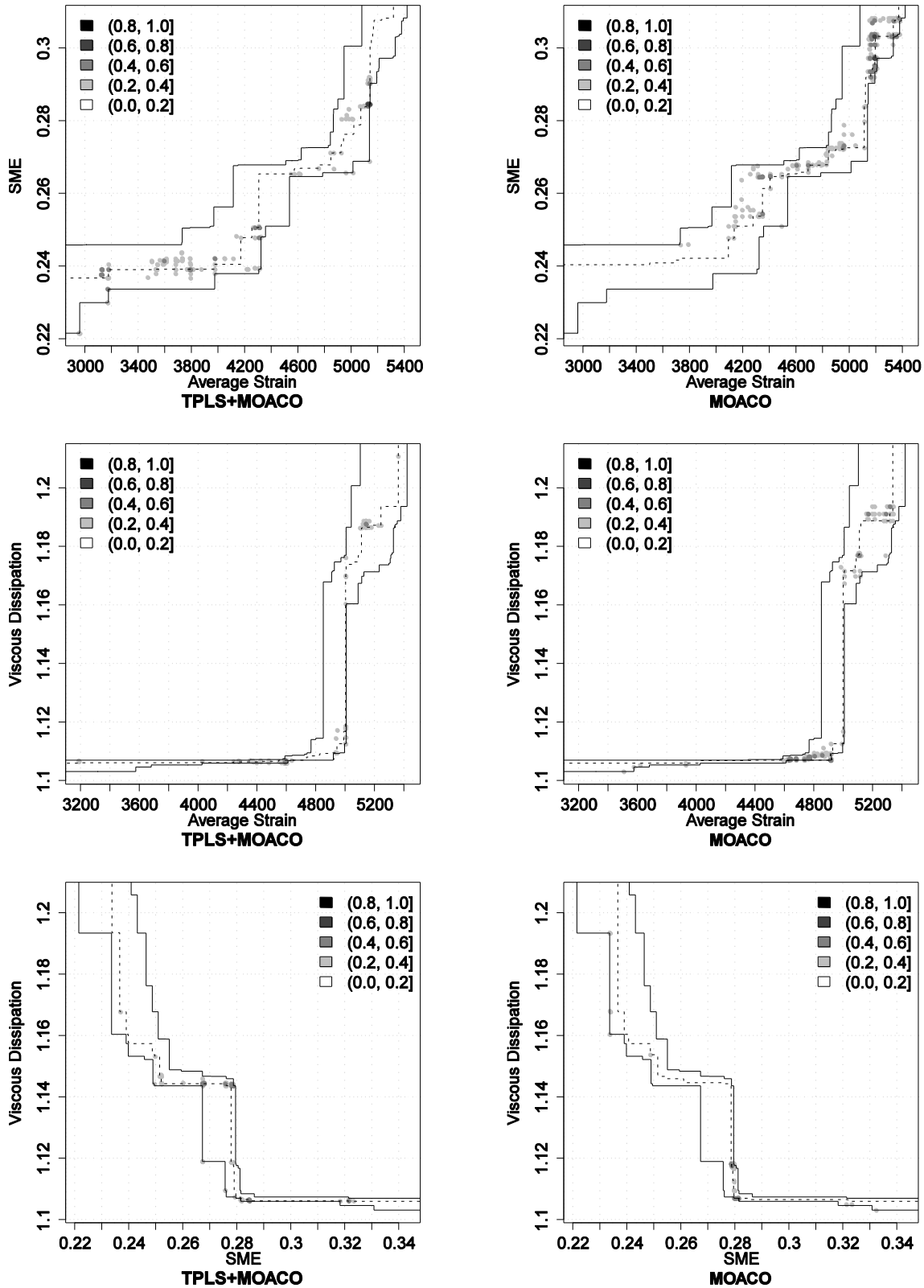

Figure C.11. Results for instance TSCP2 (see the caption of Figure C.10 for more details).

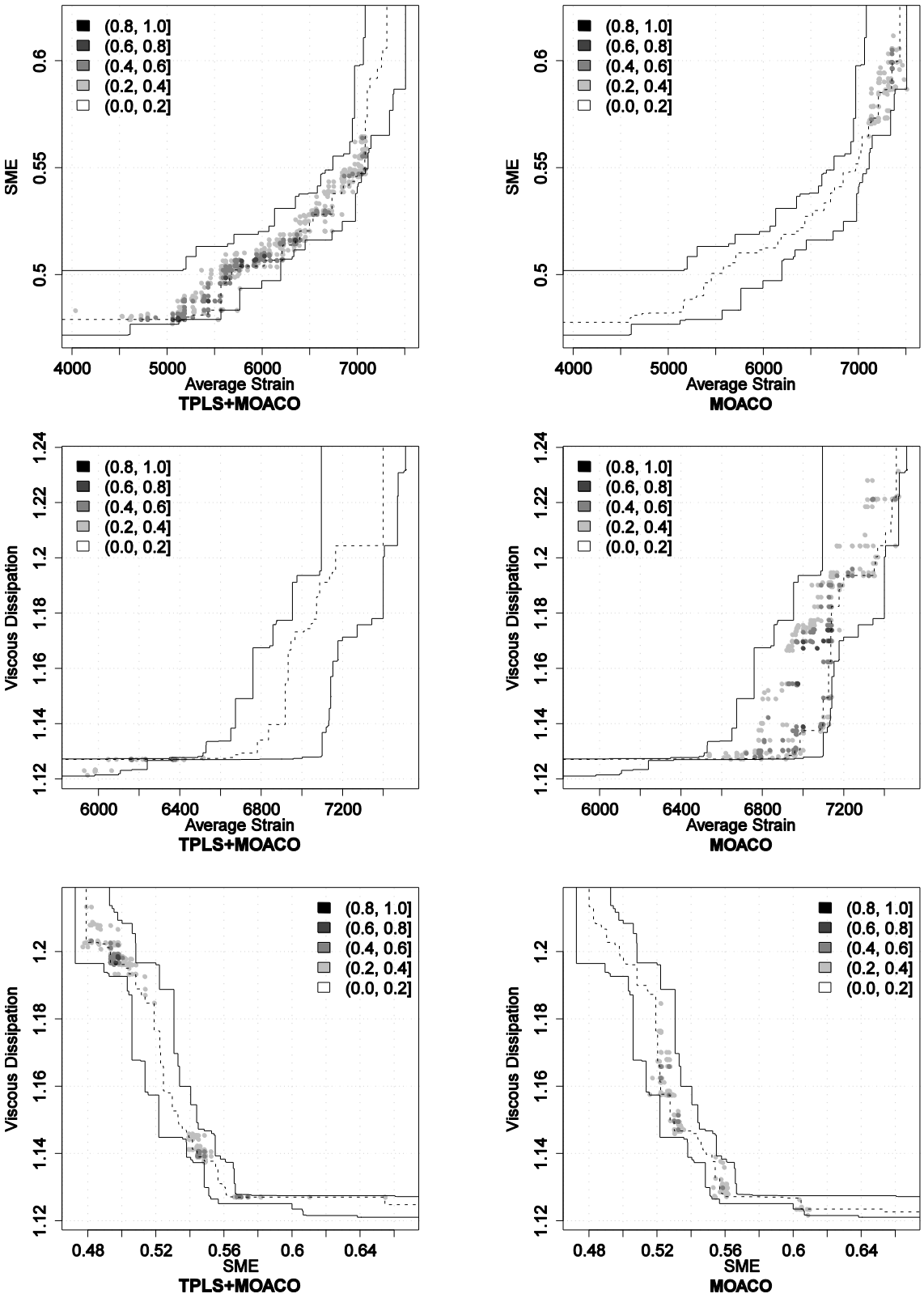

Figure C.12. Results for instance TSCP3 (see the caption of Figure C.10 for more details).

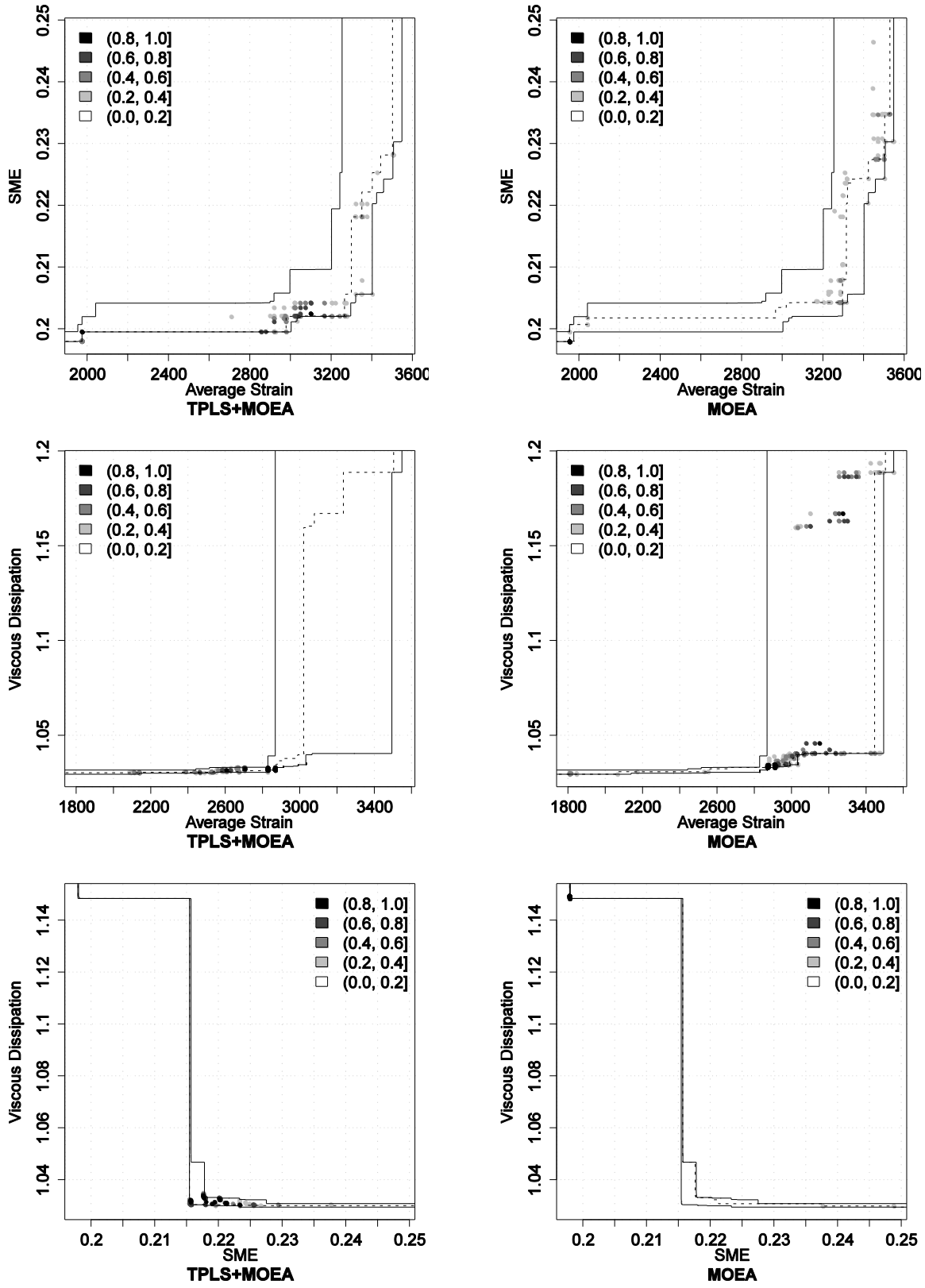

Figure C.13. Results for instance TSCP1. Differences in terms of empirical attainment functions between the TPLS+MOEA algorithm and MOEA after 3000 evaluations of the simulation program. Advantages in favor of TPLS+MOEA are indicated on the left side; those in favor of MOEA on the right side.

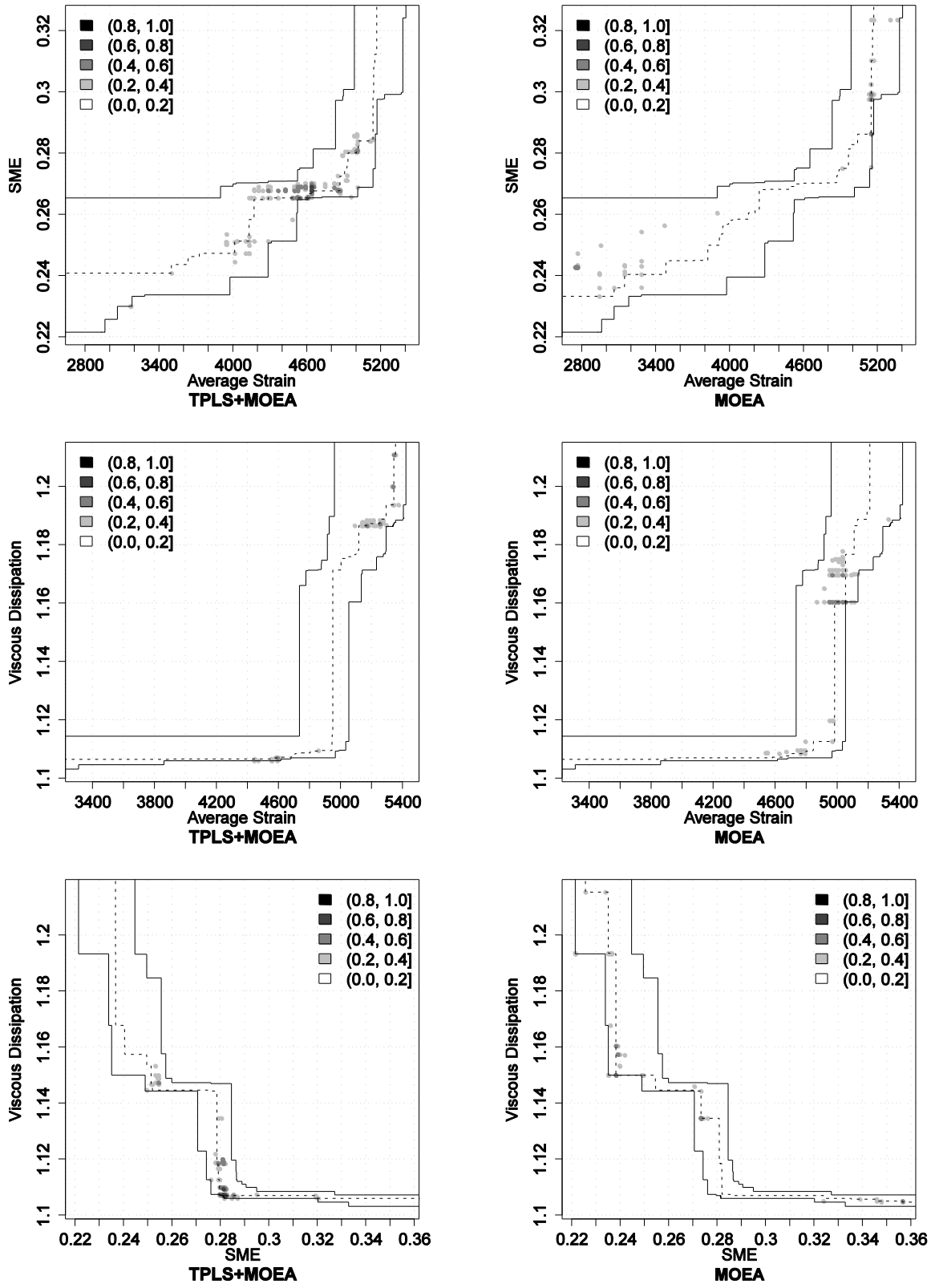

Figure C.14. Results for instance TSCP2 (see the caption of Figure C.13 for more details).

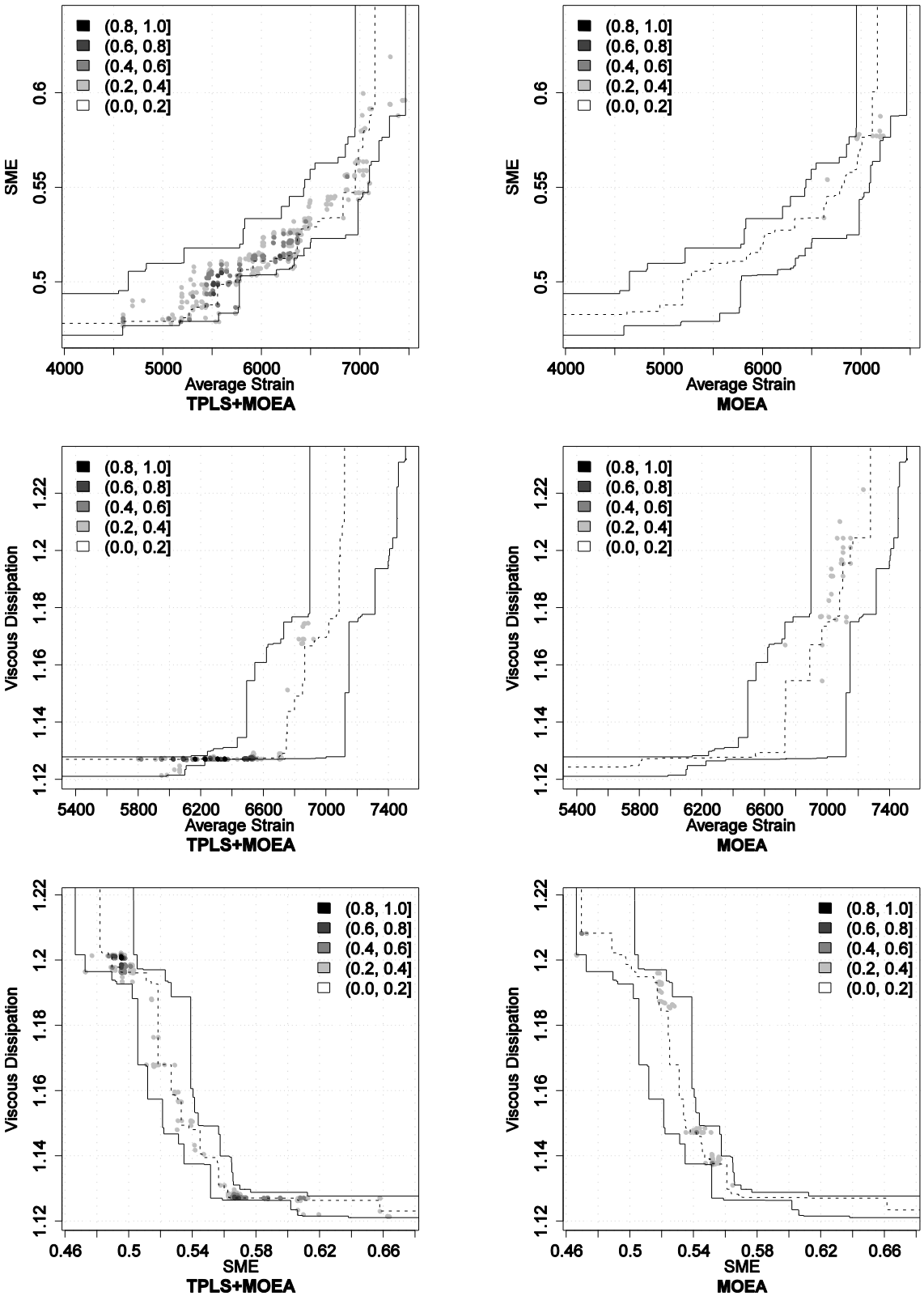

Figure C.15 Results for instance TSCP3 (see the caption of Figure C.13 for more details).

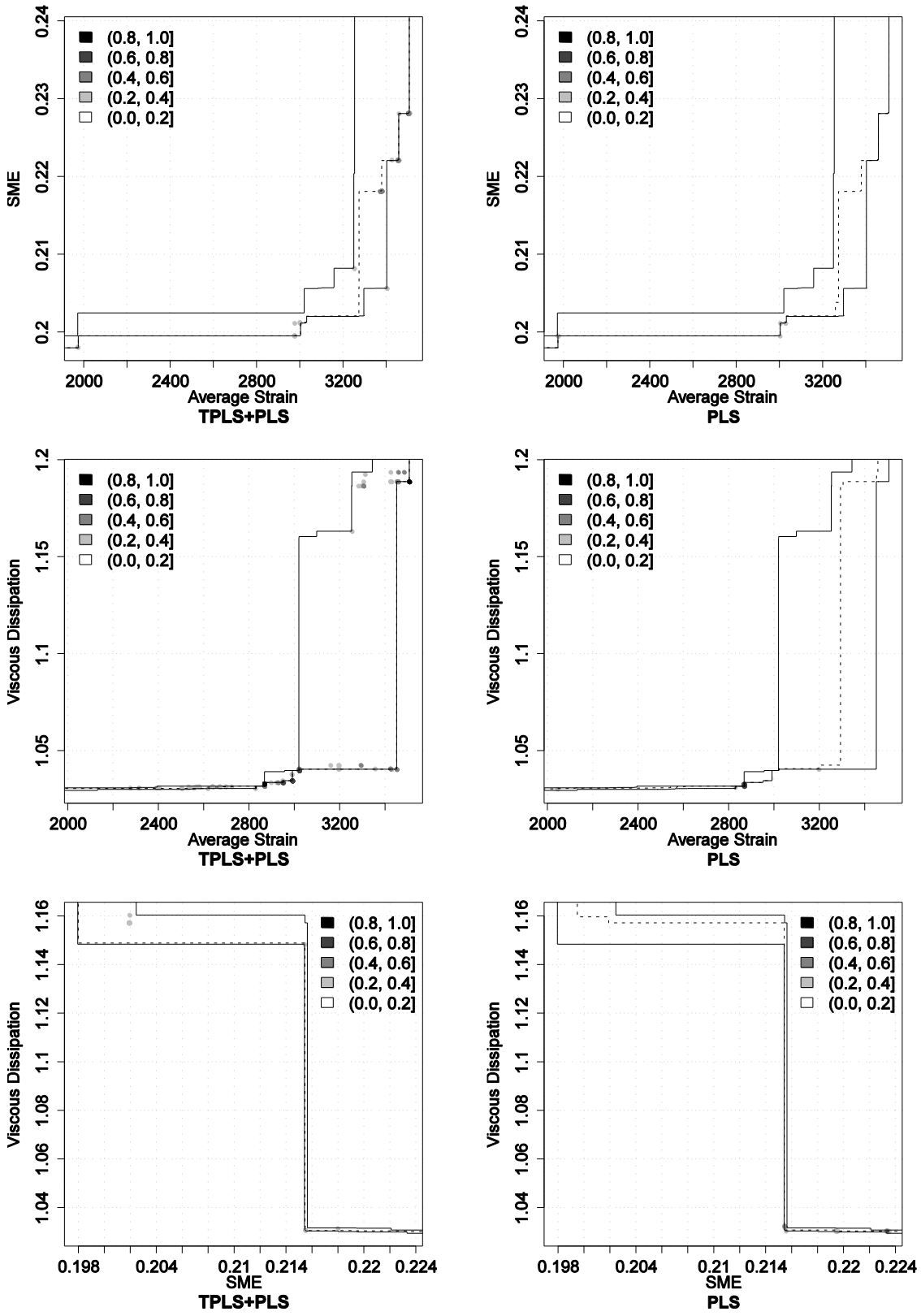

Figure C.16. Results for instance TSCP1. Differences in terms of empirical attainment functions between the TPLS+PLS algorithm and PLS after 3000 evaluations of the simulation program. Advantages in favor of TPLS+PLS are indicated on the left side; those in favor of PLS on the right side.

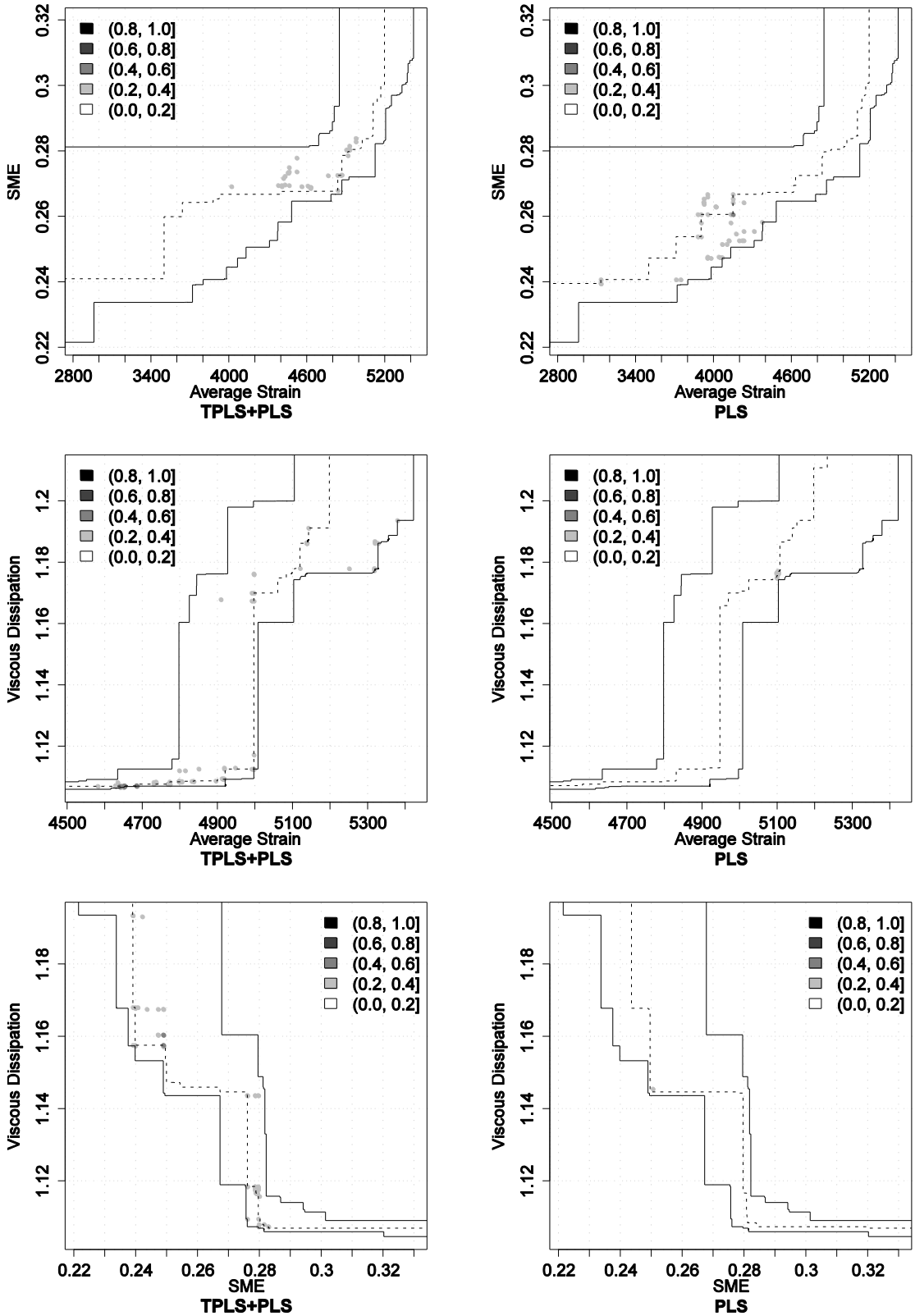

Figure C.17. Results for instance TSCP2 (see the caption of Figure C.16 for more details).

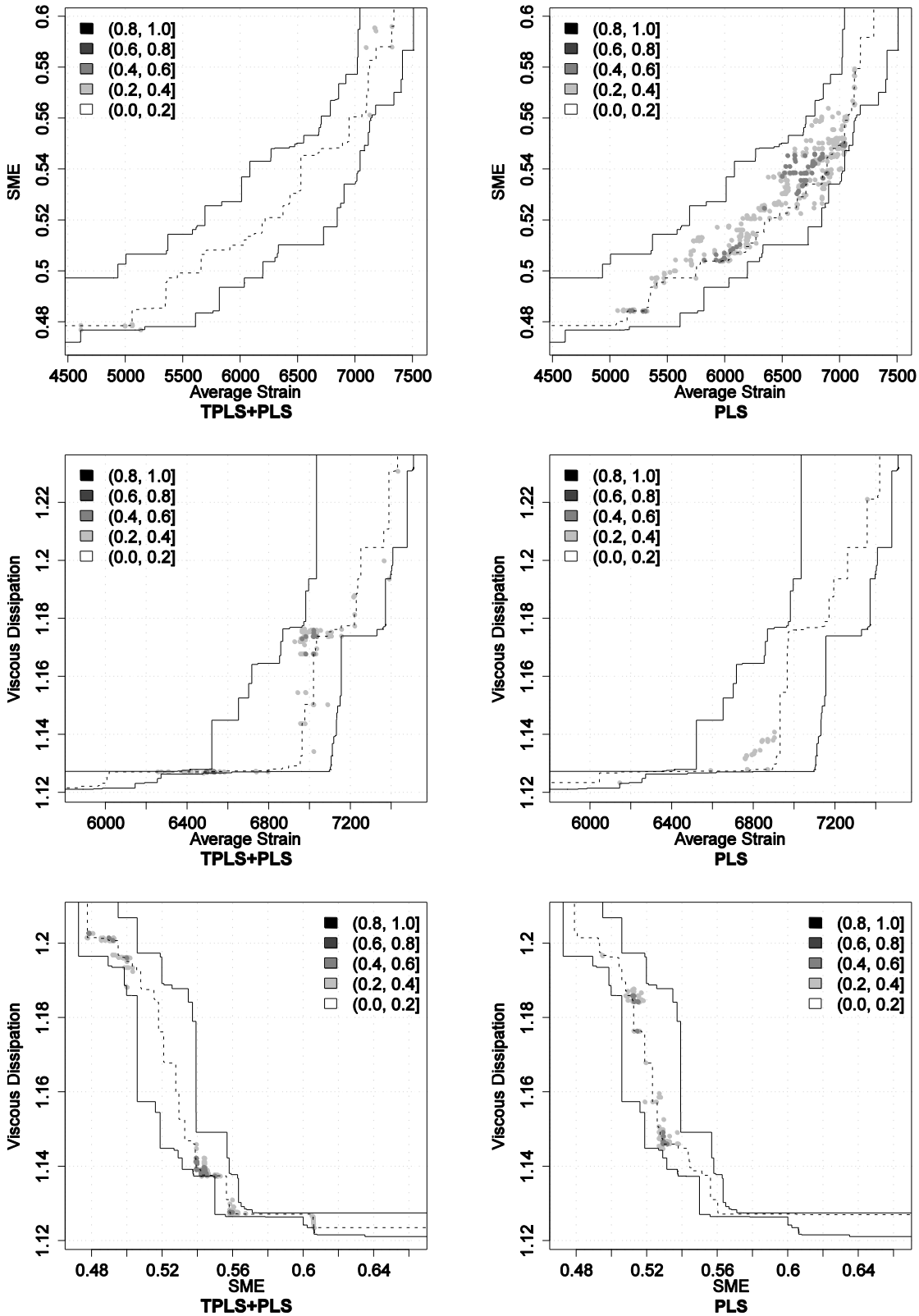

Figure C.18. Results for instance TSCP3 (see the caption of Figure C.16 for more details).

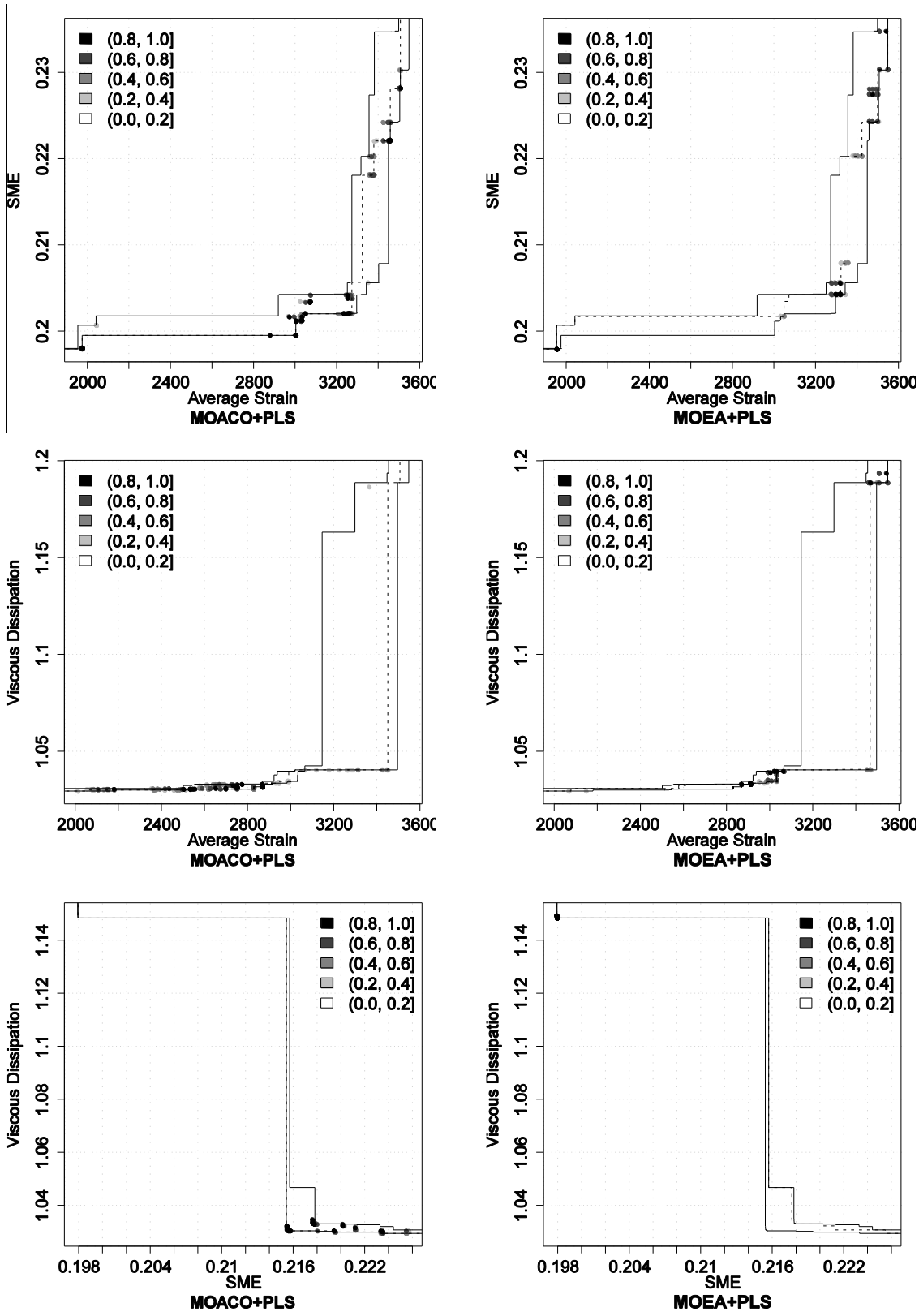

Figure C.19. Results for instance TSCP1. Differences in terms of empirical attainment functions between the MOACO+PLS algorithm and MOEA+PLS after 3000 evaluations of the simulation program. Advantages in favor of MOACO+PLS are indicated on the left side; those in favor of MOEA+PLS on the right side.

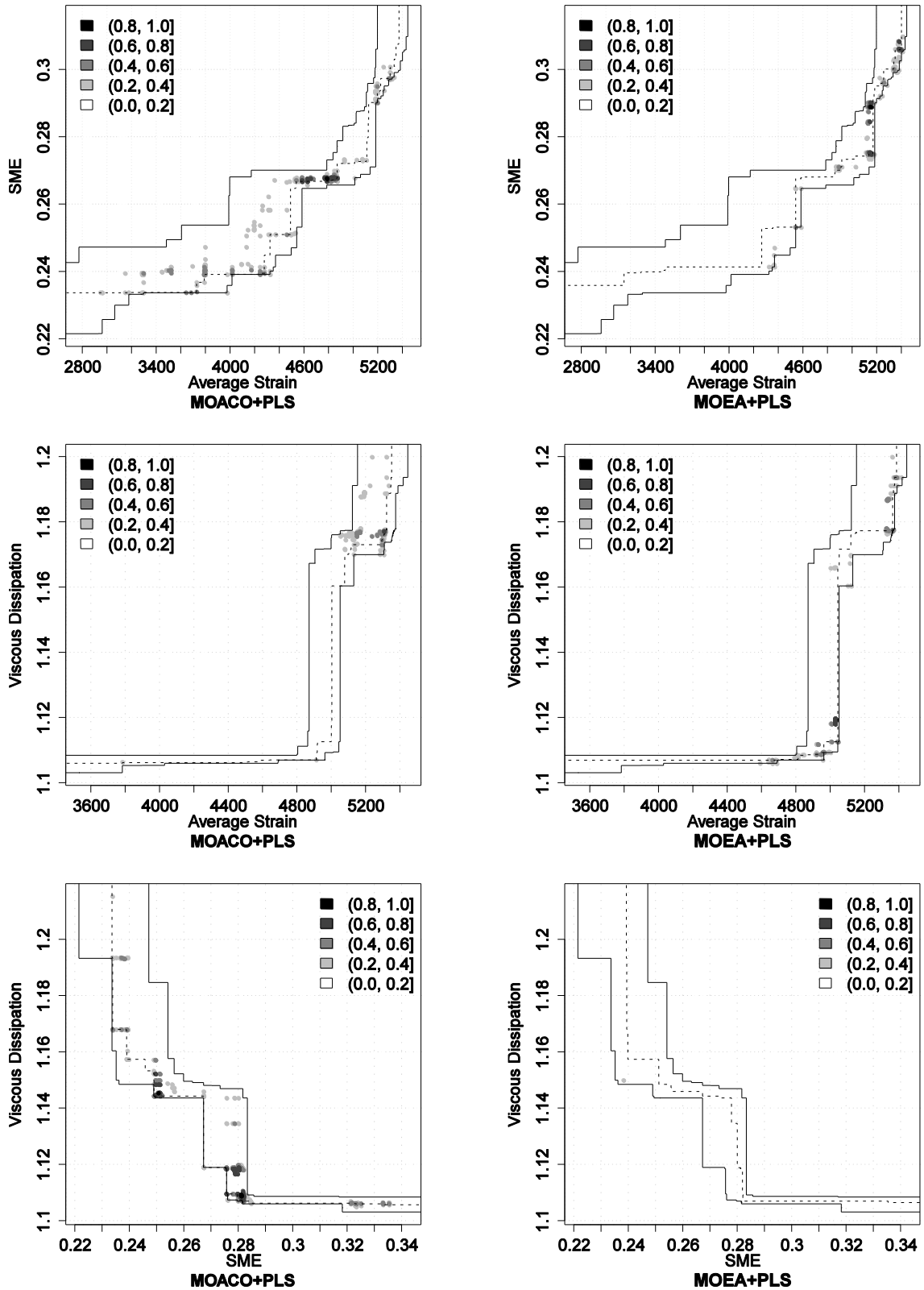

Figure C.20. Results for instance TSCP2 (see the caption of Figure C.19 for more details).

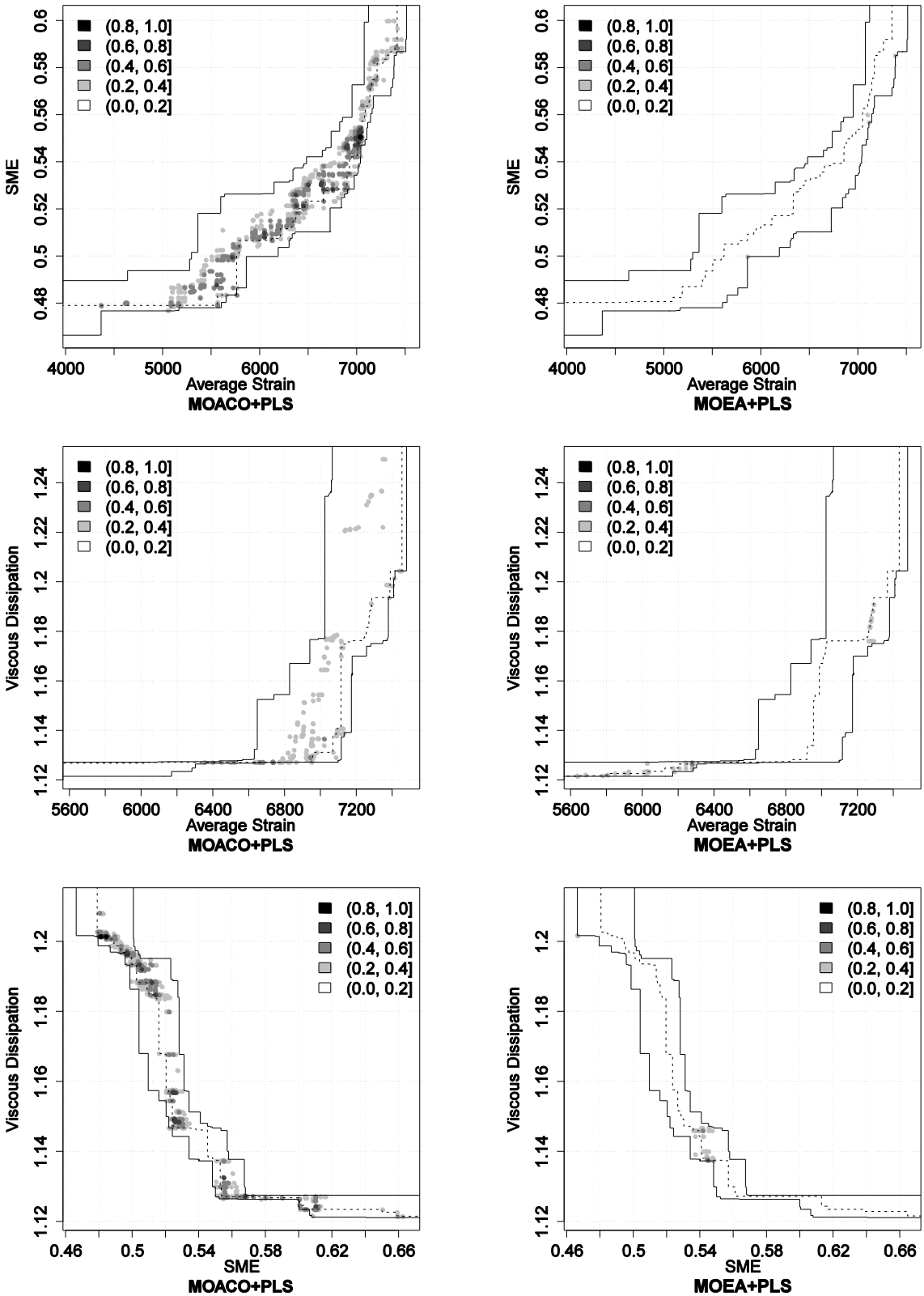

Figure C.21. Results for instance TSCP3 (see the caption of Figure C.19 for more details).

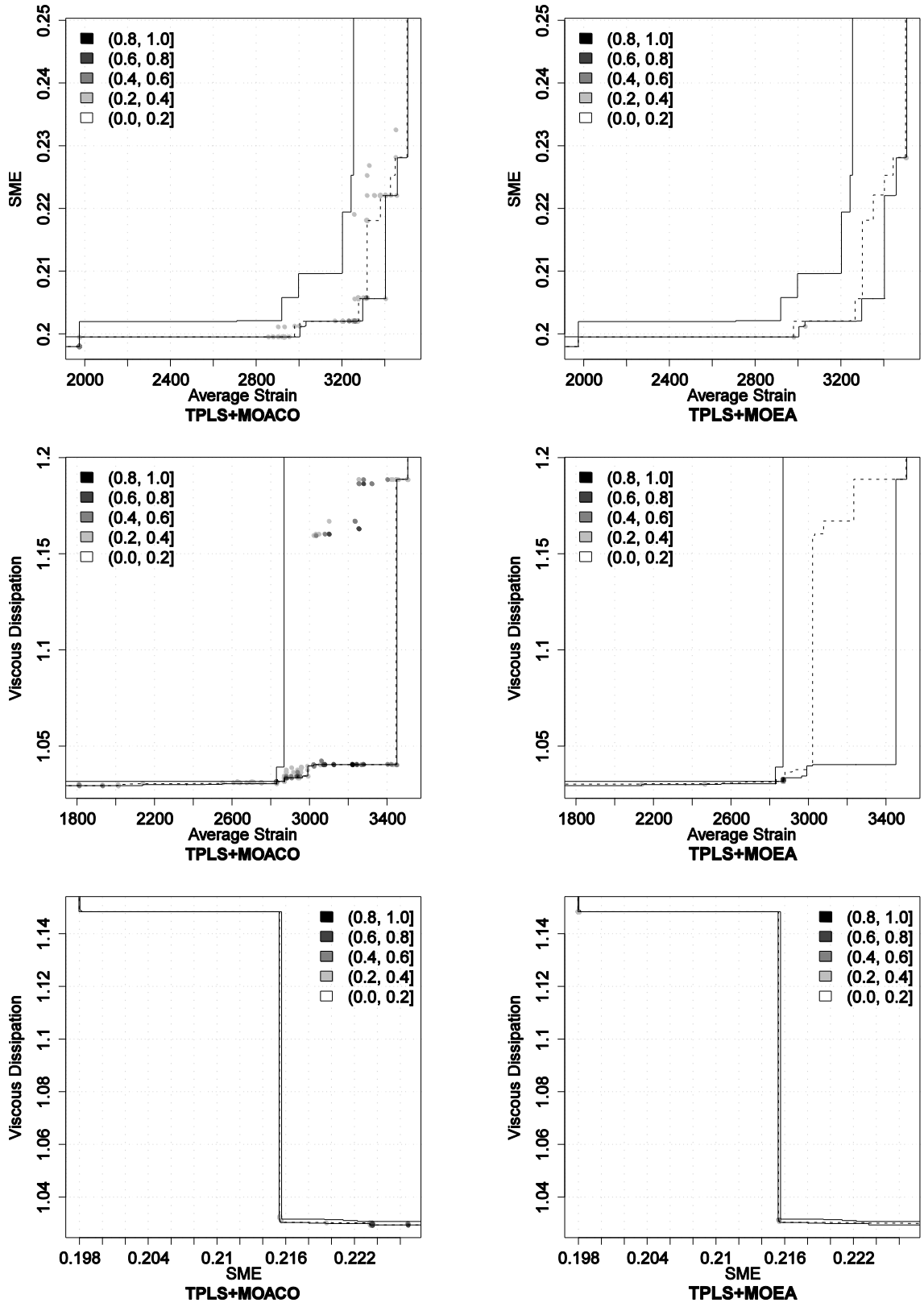

Figure C.22. Results for instance TSCP1. Differences in terms of empirical attainment functions between the TPLS+MOACO algorithm and TPLS+MOEA after 3000 evaluations of the simulation program. Advantages in favor of TPLS+MOACO are indicated on the left side; those in favor of TPLS+MOEA on the right side.

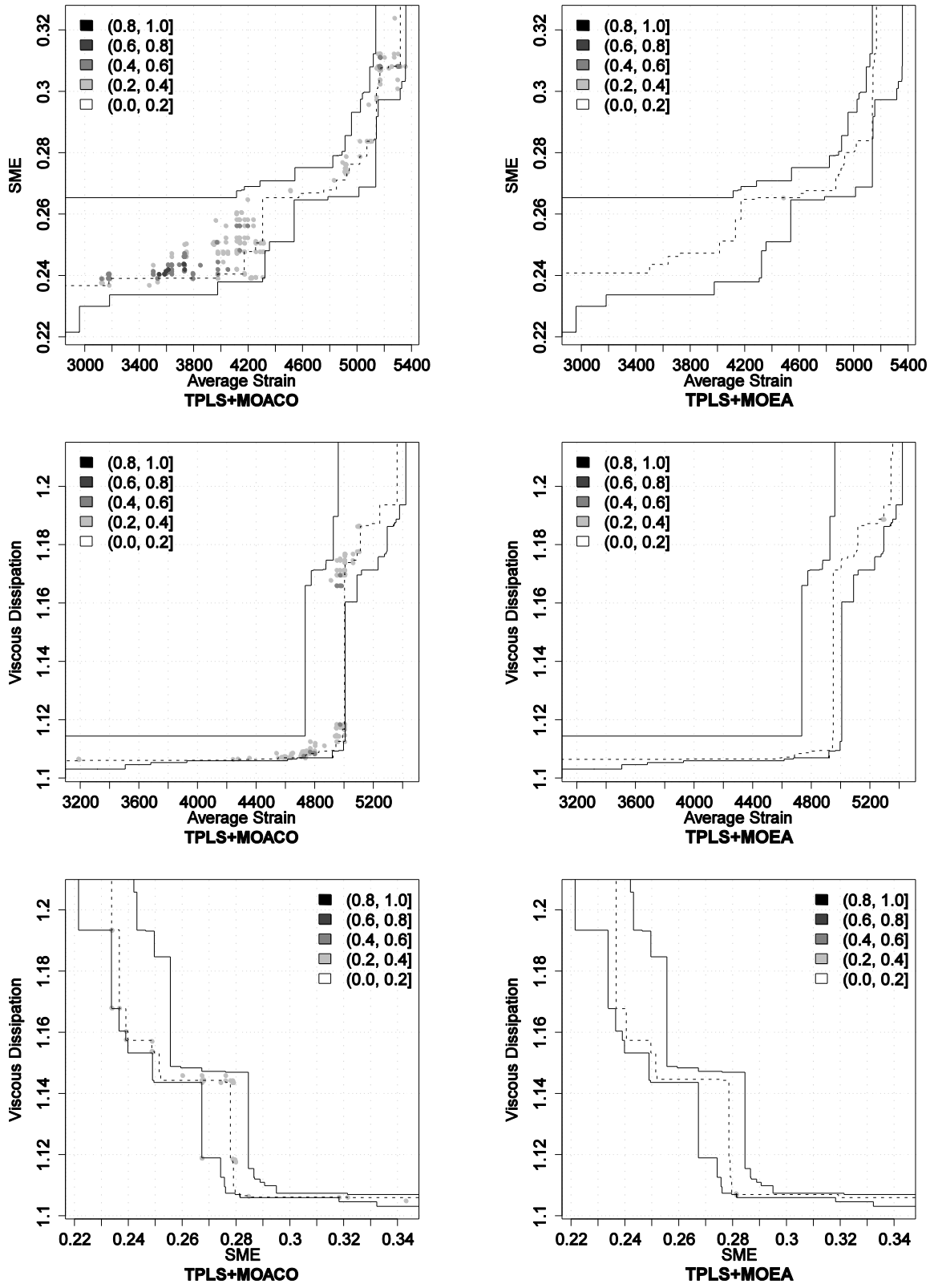

Figure C.23. Results for instance TSCP2 (see the caption of Figure C.22 for more details).

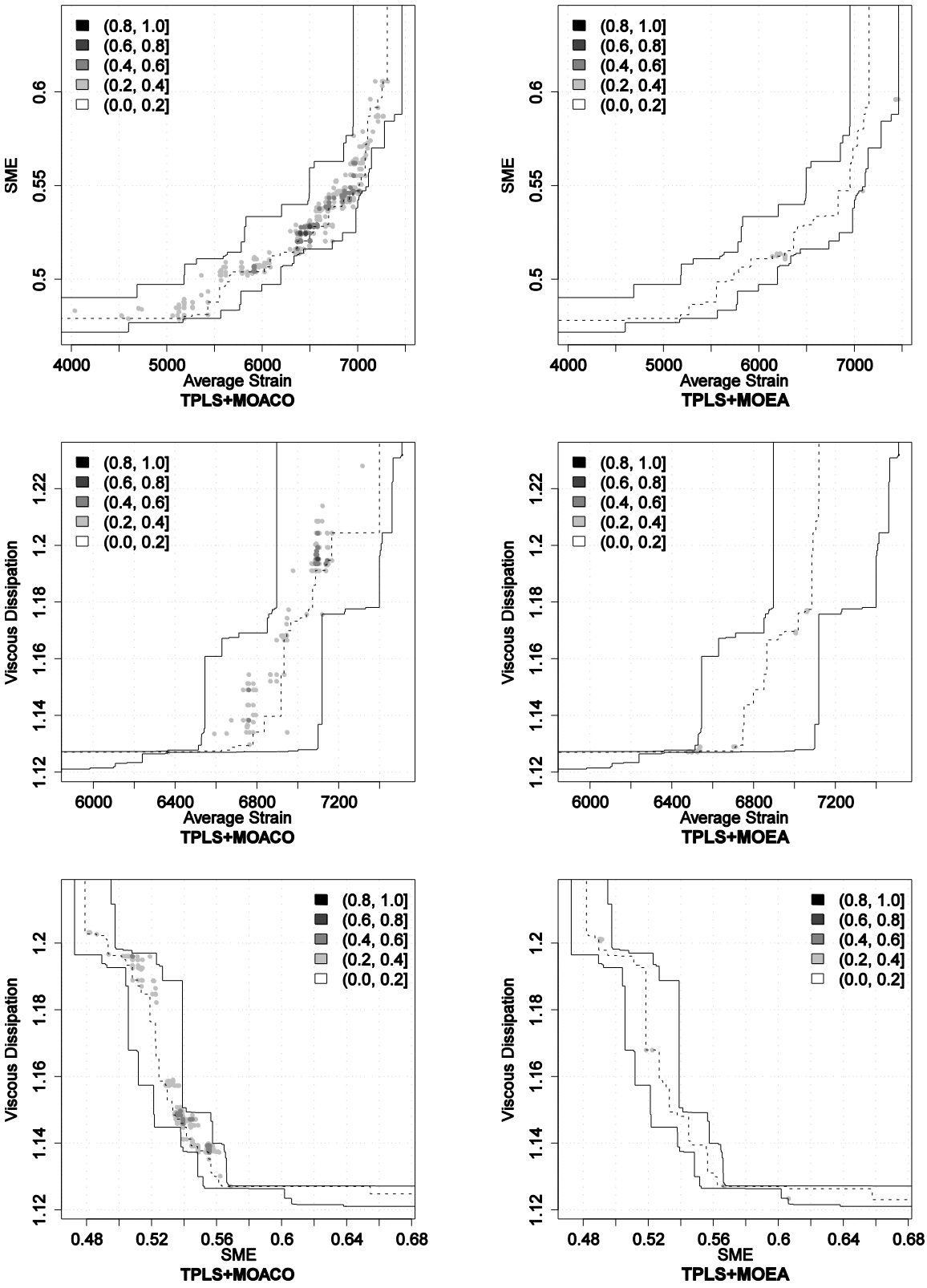

Figure C.24. Results for instance TSCP3 (see the caption of Figure C.22 for more details).

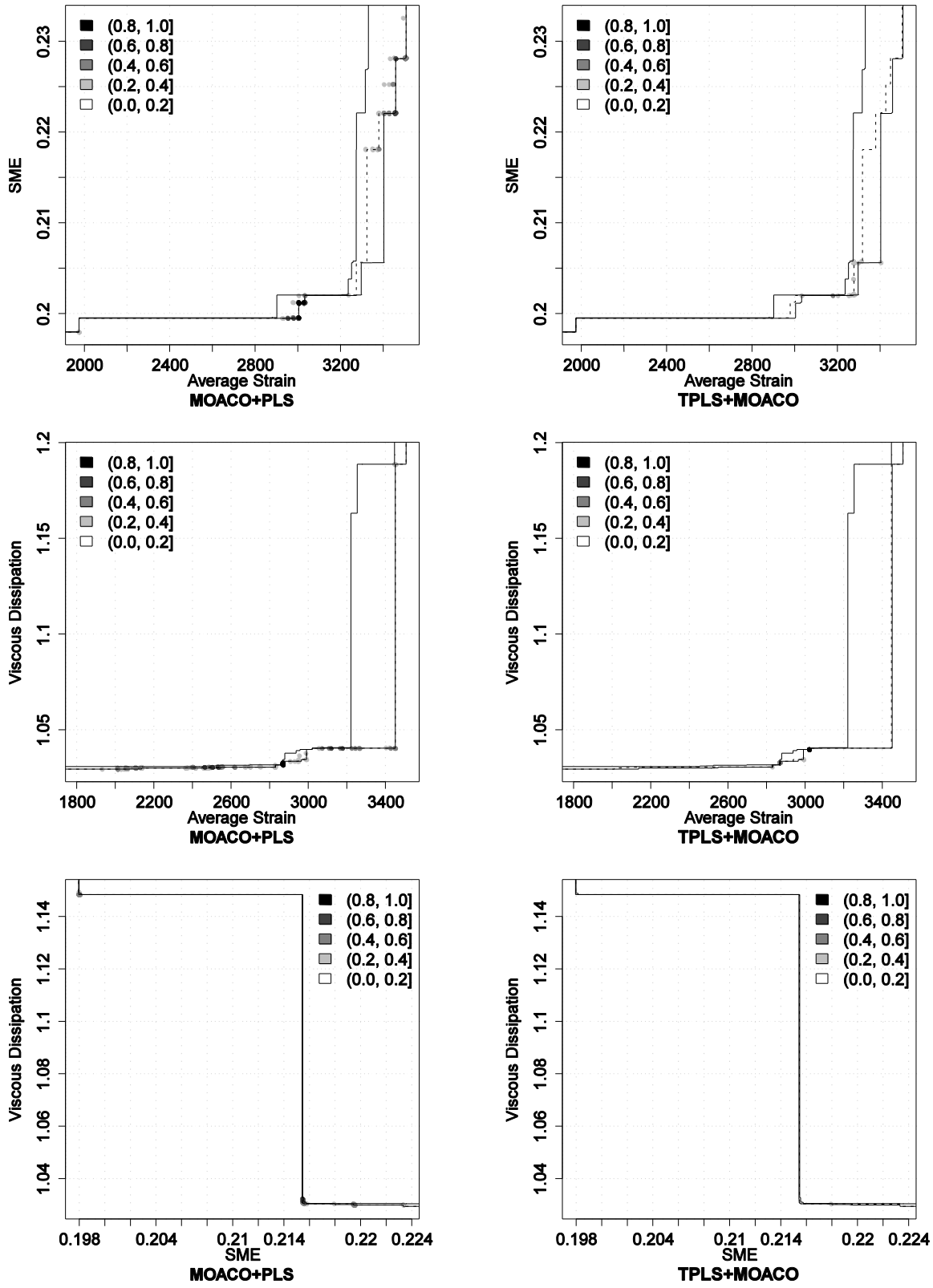

Figure C.25. Results for instance TSCP1. Differences in terms of empirical attainment functions between the MOACO+PLS algorithm and TPLS+MOACO after 3000 evaluations of the simulation program. Advantages in favor of MOACO+PLS are indicated on the left side; those in favor of TPLS+MOACO on the right side.

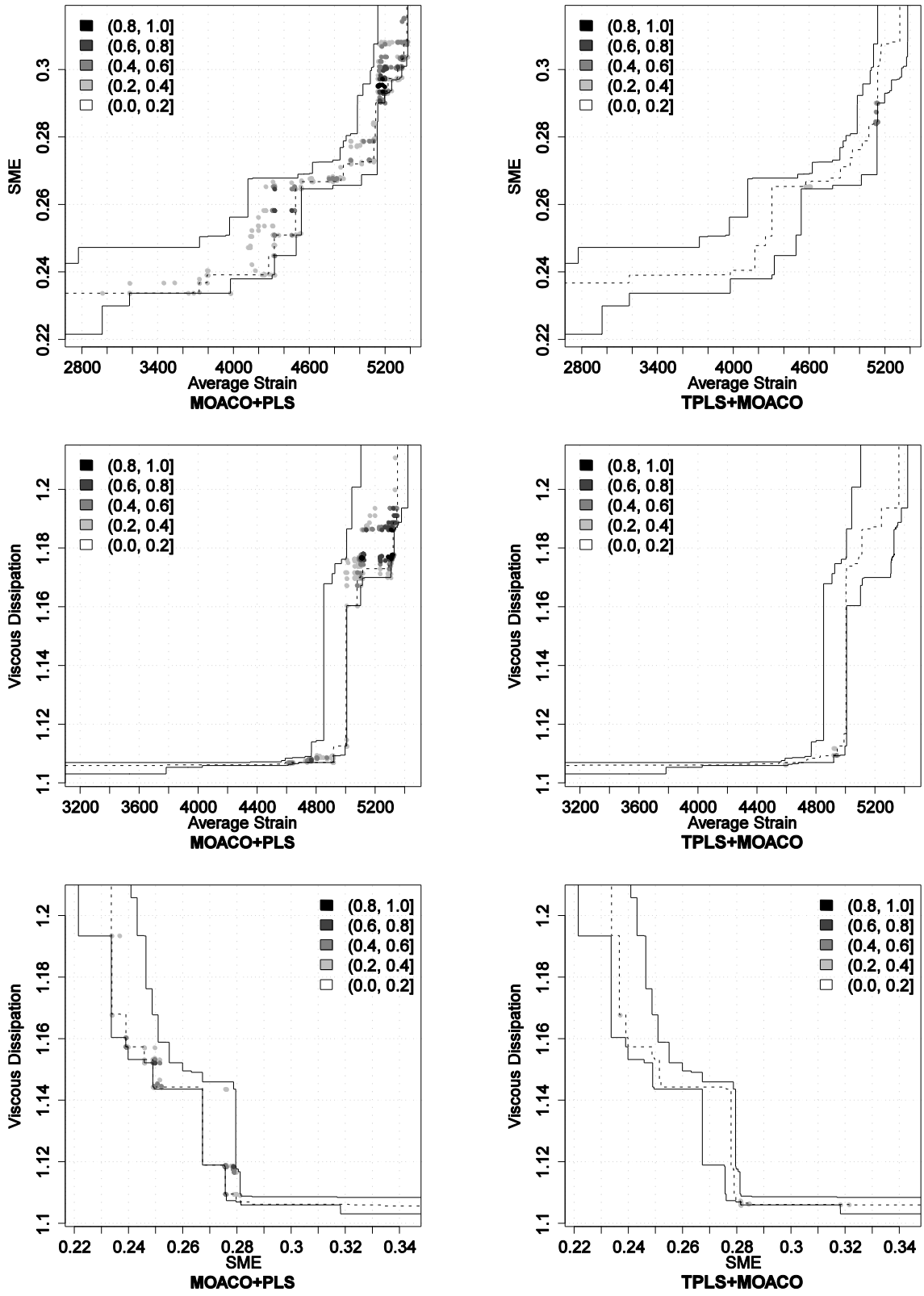

Figure C.26. Results for instance TSCP2 (see the caption of Figure C.25 for more details).

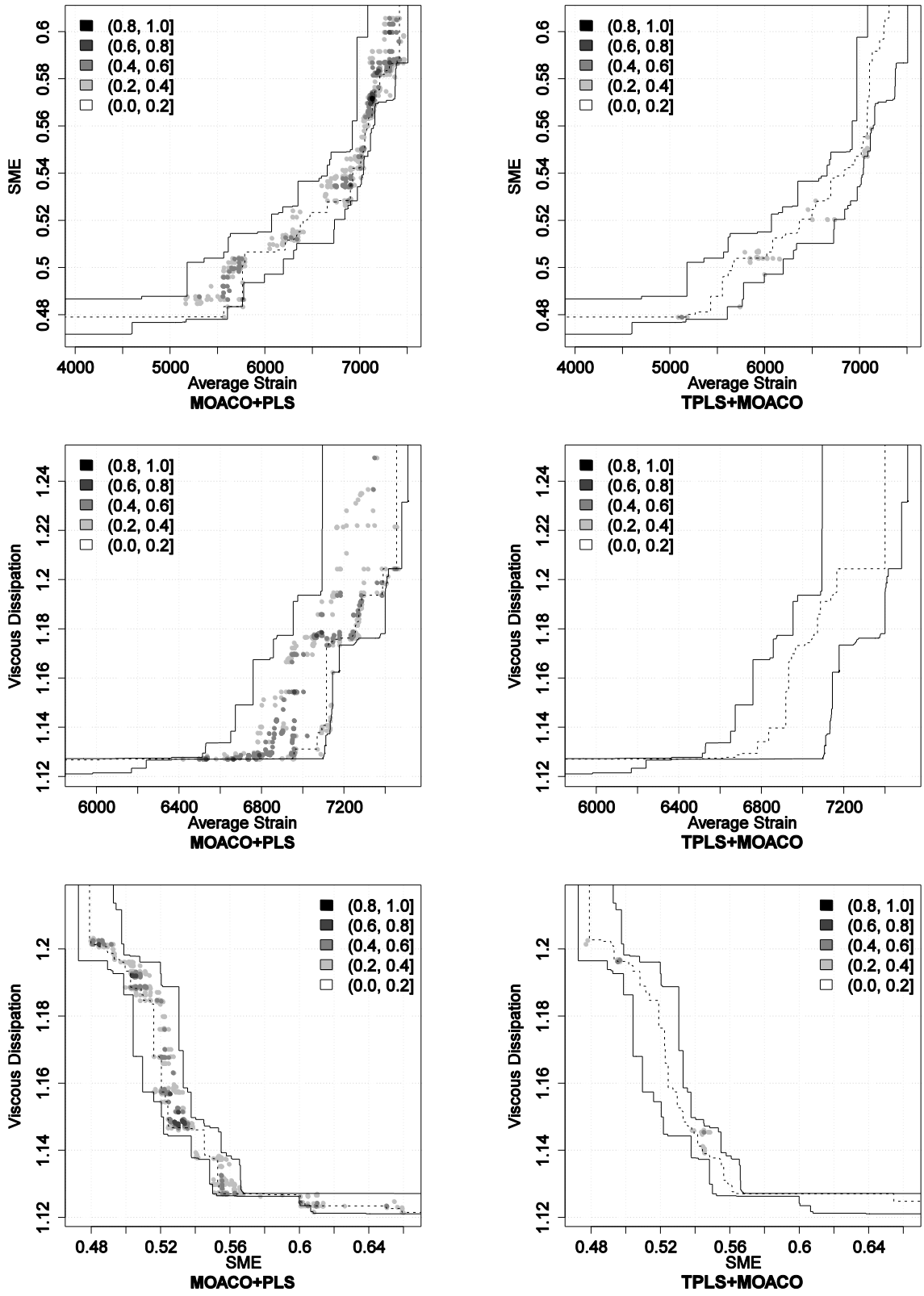

Figure C.27. Results for instance TSCP3 (see the caption of Figure C.25 for more details).

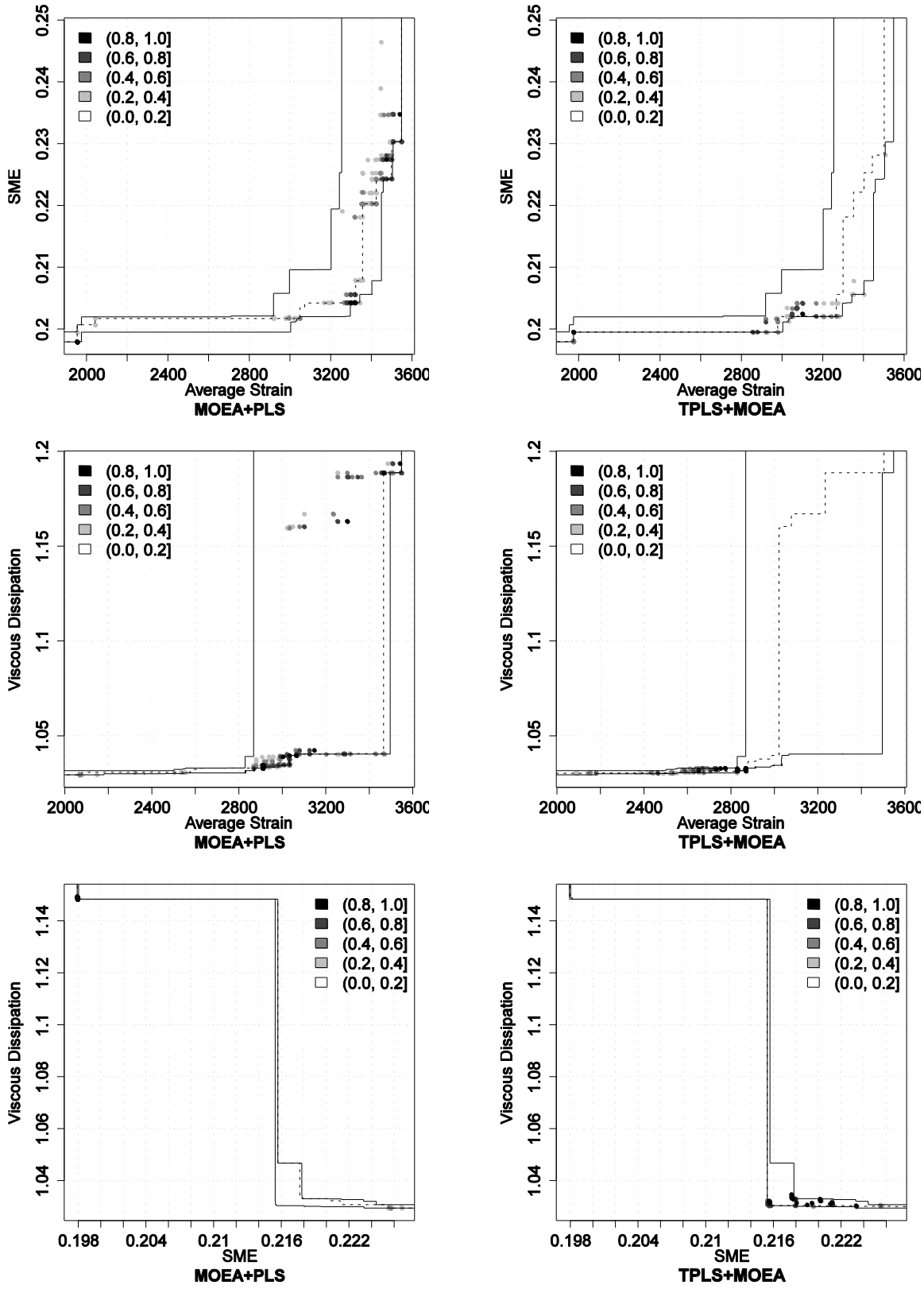

Figure C.28. Results for instance TSCP1. Differences in terms of empirical attainment functions between the MOEA+PLS algorithm and TPLS+MOEA after 3000 evaluations of the simulation program. Advantages in favor of MOEA+PLS are indicated on the left side; those in favor of TPLS+MOEA on the right side.

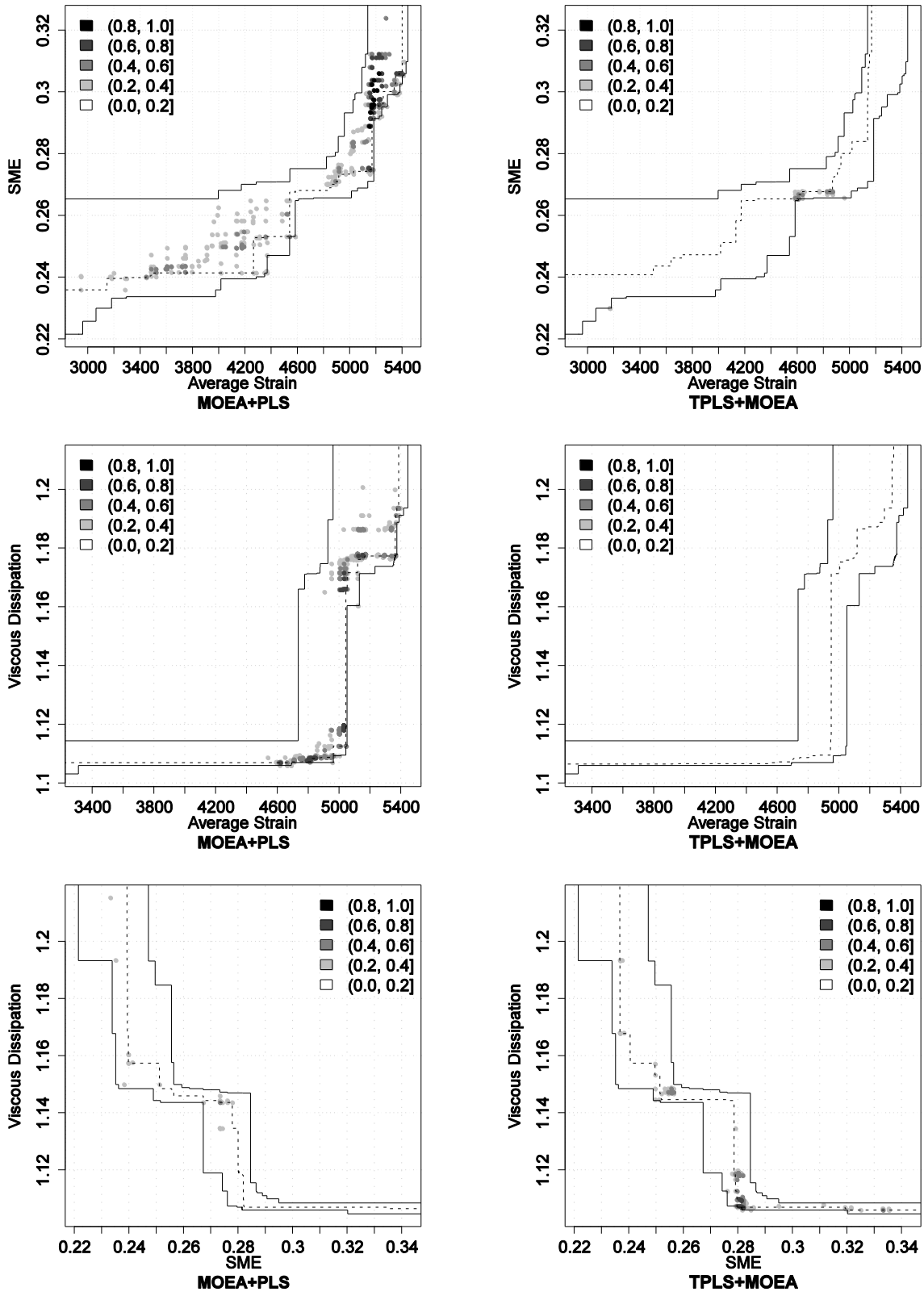

Figure C.29. Results for instance TSCP2 (see the caption of Figure C.28 for more details).

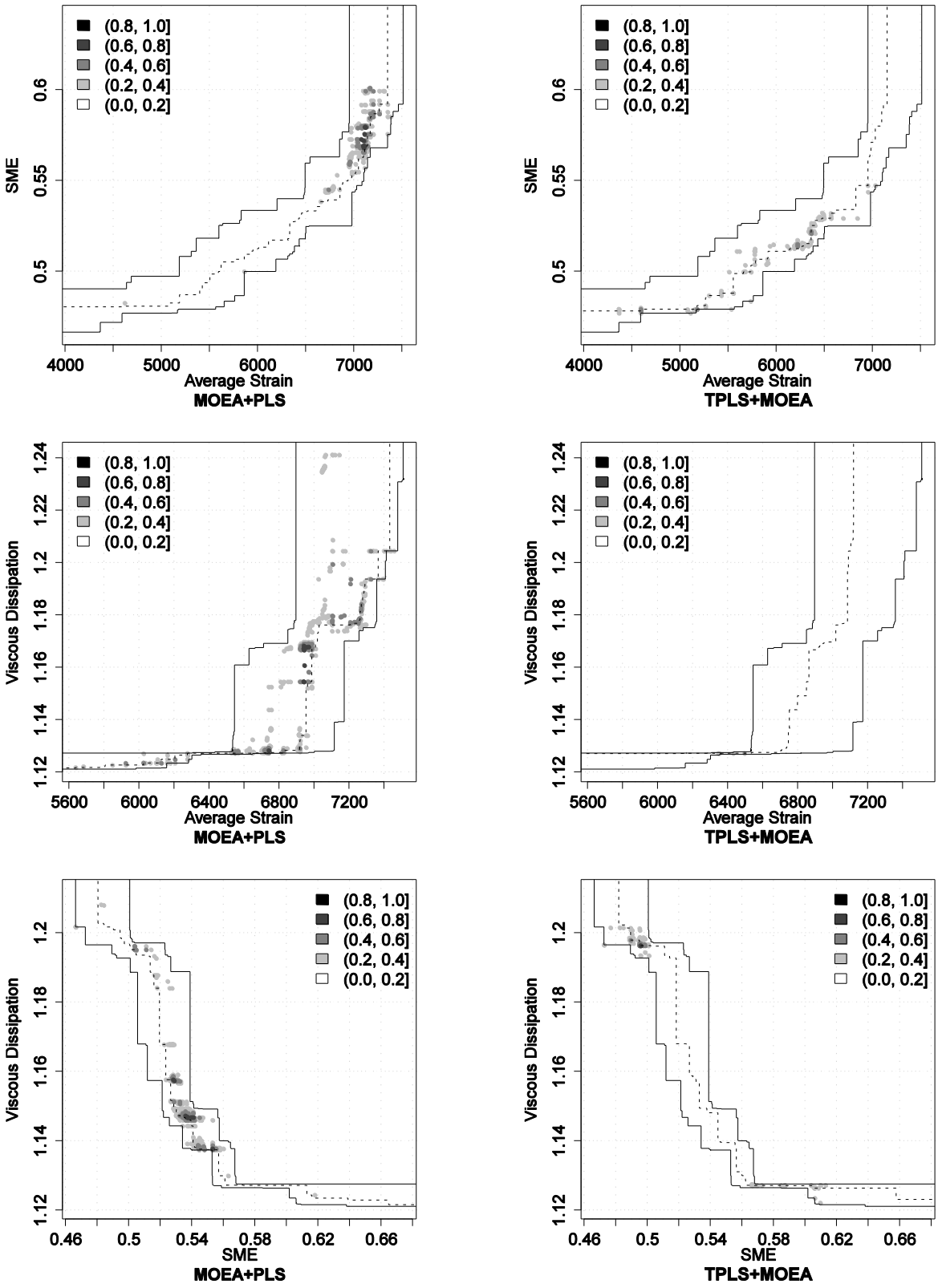

Figure C.30. Results for instance TSCP3 (see the caption of Figure C.28 for more details)